

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université M'hamed Bougara de Boumerdes

Faculté des Sciences

Département des Mathématiques



Mémoire de fin d'étude

Pour l'obtention du Diplôme de Master en Mathématiques Financières

Thème

***Etude de L'Algorithme Long Short Term Memory
(LSTM) Pour la Prédiction des Cours Boursiers***

Réalisé & présenté par

M^{lle}. Haciane Djamila & M^{me}. Griche Kaouthar

Soutenu le 06/07/2022, Devant le jury composé de :

<i>Présidente</i>	<i>M^{me}</i>	<i>METREF N.</i>	<i>M.A.A</i>	<i>U.M.B.B.</i>
<i>Examinatrice</i>	<i>M^{me}</i>	<i>DRICI W.</i>	<i>M.C.A</i>	<i>U.M.B.B.</i>
<i>Promoteur</i>	<i>M^r</i>	<i>BENAMARA O.</i>	<i>M.C.B</i>	<i>U.M.B.B.</i>
<i>Membre invité</i>	<i>M^r</i>	<i>TAHARBOUCHET S.</i>	<i>M.C.B</i>	<i>U.M.B.B.</i>

Année Universitaire 2021-2022

Remerciements

*Avant tout, on remercie **ALLAH** le tout puissant de nous avoir donné la santé, la force et le pouvoir nécessaire d'achever ce modeste travail. Qu'Allah répande ses bénédictions sur notre **prophète Mohammed**.*

On exprime notre profonde gratitude et nos sincères remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire par leur soutien moral et encouragements, et plus particulièrement :

*Notre enseignant et promoteur de mémoire, Mr **BENAMARA Oualid** pour son encadrement, sa patience, sa disponibilité, sa confiance et ses bons conseils tout le long de ce travail. Il a toujours été à l'écoute de nos nombreuses questions.*

On remercie les membres du jury d'examen pour l'honneur qu'ils nous font en participant au jugement de ce travail.

On remercie aussi tous les enseignants du département de mathématiques qui ont contribué à notre formation.

À toutes nos familles respectives qui ont toujours été là pour nous soutenir.

Enfin, nous voudrions exprimer nos remerciements à toute personne qui a contribué de près ou de loin pour la finalisation de ce travail.



Merci

Dédicaces



Avant tout, ma plus grande reconnaissance est adressée à Dieu, tout-puissant de m'avoir donné le courage, la volonté et la santé pour achever ce travail.

Dans ma vie j'étais toujours reconnaissante pour des personnes qui comptent beaucoup pour moi. Aujourd'hui l'occasion est venue pour leur offrir ce travail en signe de reconnaissance pour tout ce qu'ils ont consenti d'efforts rien que pour me voir réussir.

Je dédie ce travail :

À mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études. Je vous souhaite une vie pleine de bonheur et de santé.

À mon frère & ma sœur, pour leur appui et leur encouragement.

À mes deux grands-mères adorées pour toute leur tendresse et leur soutien moral.

À la mémoire de mes deux grands-pères Saïd & Ahmed, que j'aimerais pour toujours.

À mes tantes & mes oncles, mes cousines & cousins.

À toute ma famille sans exception.

À mon cher binôme Kaouthar, je voudrais te remercier pour les beaux moments que nous avons passés ensemble pour accomplir ce travail.



Djamila

Dédicaces



Avant tout, ma plus grande reconnaissance est adressée à Dieu, tout-puissant de m'avoir donné le courage, la volonté et la santé pour achever ce travail.

Dans ma vie j'étais toujours reconnaissante pour des personnes qui comptent beaucoup pour moi. Aujourd'hui l'occasion est venue pour leur offrir ce travail en signe de reconnaissance pour tout ce qu'ils ont consenti d'efforts rien que pour me voir réussir.

Je dédie ce travail :

À mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études, je vous souhaite une vie pleine de bonheur et de santé.

À mon cher mari, merci de m'avoir soutenu, tu m'as toujours encouragé et inspiré pour faire ce travail, je ne te remercierai jamais assez.

À mes frères & sœurs, pour leur appui et leur encouragement.

À mes grands-parents maternels adorés pour toute leur tendresse et leur soutien moral.

À la mémoire de mes grands-parents paternels, que j'aimerais pour toujours.

À mes tantes & mes oncles, mes cousines & cousins.

À toute ma famille sans exception.

À mon cher binôme Djamila, je voudrais te remercier pour les beaux moments que nous avons passés ensemble pour accomplir ce travail.



Kacouther

Table des matières

Table des matières	V
Table des figures	IX
Liste des tableaux	XI
Notions de base	XII
Introduction Générale	1
1 Le marché boursier	3
1.1 Introduction	4
1.2 Histoire rapide	4
1.3 Définition	5
1.3.1 Marché primaire	5
1.3.2 Marché secondaire	6
1.4 La loi de l'offre et de la demande	6
1.5 Les titres de financement les plus courants sur le marché boursier	7
1.5.1 Les actions	7
1.5.2 Les obligations	8
1.6 Rôle et fonctionnement	10
1.6.1 Rôle	10
1.6.2 Fonctionnement	11
1.7 Les ETF	12
1.8 La prédiction boursière	14
1.9 Conclusion	15
2 Machine Learning	16
2.1 Introduction	17
2.2 Historique	17
2.3 Définition	18
2.4 Les notions clés du Machine Learning	18
2.4.1 Machine Learning « Step by Step »	19

2.5	BigData & Deep Learning	19
2.5.1	Machine Learning et BigData deux notions indissociables.....	19
2.5.2	Machine Learning vs Deep Learning	20
2.6	Les principaux algorithmes de Machine Learning.....	20
2.6.1	Apprentissage supervisé	20
2.6.2	Apprentissage non supervisé	21
2.6.3	Apprentissage par renforcement	22
2.6.4	Apprentissage Semi-supervisé.....	23
2.6.5	Récapitulation	24
2.7	Modèle de Machine Learning	25
2.7.1	Définition.....	25
2.7.2	Développement d'un modèle de Machine Learning	25
2.8	Fonctionnement et domaine d'application.....	25
2.8.1	Pourquoi le Machine Learning est utilisé.....	25
2.8.2	Fonctionnement.....	26
2.8.3	Domaine d'application	27
2.8.4	Dans le monde de la finance.....	27
2.8.5	Analyse prédictive.....	27
2.9	Conclusion	28
3	Les Réseaux De Neurones Artificiel.....	29
3.1	Introduction.....	30
3.2	Historique.....	30
3.3	Du neurone biologique au neurone artificiel	32
3.3.1	Analogie avec le cerveau.....	32
3.3.2	Aspects historiques.....	32
3.3.3	Définition.....	33
3.3.4	Structure des neurones.....	33
3.4	Neurones artificiels.....	34
3.4.1	Définition.....	34
3.4.2	Comportement.....	35
3.4.3	Fonction d'activation.....	35
3.4.4	L'architecture des réseaux de neurones	36
3.4.4.1	Réseaux de neurones non bouclés	37

3.4.4.2	Réseaux de neurones bouclés	38
3.5	Réseaux de neurones multicouche	39
3.5.1	Définition.....	40
3.5.2	Fonctionnement.....	40
3.6	Le neurone sigmoïde	41
3.6.1	Définition.....	41
3.6.2	Fonctionnement.....	41
3.6.3	Comme fonction d'activation dans les réseaux de neurones	43
3.6.4	L'importance de la fonction sigmoïde dans les réseaux de neurones	44
3.7	Les règles d'apprentissage des réseaux de neurones.....	44
3.7.1	Règle de Hebb.....	45
3.7.2	Règle de Widrow Hoff	46
3.8	L'algorithme d'apprentissage standard pour les réseaux de neurones.....	46
3.8.1	L'algorithme back propagation du gradient	50
3.9	Conclusion	52
4	La mémoire longue à court terme « LSTM »	53
4.1	Introduction.....	54
4.2	Réseaux de neurones récurrents	54
4.2.1	Définition.....	54
4.2.2	Représentation.....	56
4.2.3	Exemples d'application RNN.....	58
4.3	Le problème du gradient de fuite	59
4.3.1	Bref historique.....	59
4.3.2	Expliquer le problème du gradient de fuite	59
4.3.3	Qu'est-ce que cela signifie pour le réseau ?.....	60
4.3.4	Solutions au problème du gradient de fuite	61
4.3.5	Problème du gradient de fuite et le LSTM	61
4.4	Mémoire à long court terme (LSTM).....	62
4.4.1	Architecture LSTM	62
4.4.2	Procédure pas à pas de LSTM	63
4.4.3	Variante LSTM.....	66
4.5	Conclusion	68

5	Application.....	69
5.1	Introduction.....	70
5.2	Préparation pour exécution	70
5.3	Présentation des résultats des programmes.....	71
5.4	Conclusion	76
	 Conclusion générale.....	 77
	 Annexe pour les algorithmes.....	 78
	 Bibliographie.....	 84

Table des figures

Figure 1-1 : Création de la premier bourse.	4
Figure 1-2 : Schéma d'un marché dématérialisé.	5
Figure 1-3 : Courbe de l'offre et la demande avec concurrence pure et parfaite.	7
Figure 1-4 : Ventes aux enchères.	11
Figure 1-5 : La bourse virtuelle.	11
Figure 1-6 : Exemple de la reproduction de l'évolution de la valeur de L'EURO STOXX 50.	13
Figure 2-1 : Classement des concepts	19
Figure 2-2 : Les principaux algorithmes du Machine Learning.	24
Figure 2-3 : Fonction $F(X)=AX+B$	26
Figure 3-1 : Représentation d'un neurone biologique.	33
Figure 3-2 : Modèle de neurones artificiels de Mac Culloch et Pitts.	34
Figure 3-3 : Les fonctions d'Activation d'un réseau de neurone artificiel.	36
Figure 3-4: Architecture d'un réseau de neurones à N entrées, une couche de N_c neurones caches et M neurones de sortie.	37
Figure 3-5 : Connexions : (a) partielles et (b) totales.	38
Figure 3-6 : Connexions (a) directes et (b) récurrentes.	38
Figure 3-7: Réseaux de neurones a une seule couche « le perceptron simple»	39
Figure 3-8 : Réseaux de neurones multicouche « le perceptron multicouche»	39
Figure 3-9 : Réseaux de neurones multicouche «le perceptron avec quatre couches»	40
Figure 3-10 : Représentation de la fonction sigmoïde.	42
Figure 3-11 : Représentation de la fonction en escalier	42
Figure 3-12 : Fonction d'activation dans une couche d'un réseau de neurones	43
Figure 3-13 : Exemple d'une représentation de réseau de neurone	48
Figure 3-14 : Organisme de l'algorithme de rétro-propagation du gradient.	51
Figure 3-15 : Schéma de fonctionnement de l'algorithme de rétro propagation	52

Figure 4-1 : Réseau de neurones récurrents.....	55
Figure 4-2 : Représentation N°1	56
Figure 4-3 : Représentation N°2	57
Figure 4-4 : Représentation N°3	57
Figure 4-5 : Représentation N°4	57
Figure 4-6 : Représentation N°5	58
Figure 4-7 : Représentation de cinq exemples d'application RNN.....	58
Figure 4-8 : Le voyage des observations dans un RNN	59
Figure 4-9 : Les résultats du voyage des observations dans les RNN.....	60
Figure 4-10 : Module répétitif dans un RNN standard contient une seule couche	62
Figure 4-11 : Module répétitif dans un LSTM standard contient quatre couches	63
Figure 4-12 : Représentation plus approchée des continus de la Figure 4-11	63
Figure 4-13 : Porte d'oubli.....	64
Figure 4-14 : Mise à jour de la cellule.....	64
Figure 4-15 : Mise à jour de l'ancien état de cellule	65
Figure 4-16 : La sortie de réseau LSTM.....	65
Figure 4-17 : Un LSTM standard.....	66
Figure 4-18 : La variante n°1 du LSTM.....	67
Figure 4-19 : La variante n°2 du LSTM.....	67
Figure 4-20 : La variante n°3 du LSTM.....	68
Figure 5-1 : Prédiction du cours de blé.....	71
Figure 5-2 : Prédiction du cours de l'or.....	72
Figure 5-3 : Prédiction du cours de pétrole.....	73
Figure 5-4 : Prédiction du cours de l'action Google	74
Figure 5-5 : Prédiction du cours de l'action Boeing	75

Liste des tableaux

Tableau 3-1 : Equivalence entre le neurone biologique et le neurone artificiel.	36
Tableau 3-2 : Résumé de la rétro-propagation du gradient	47
Tableau 5-1 : Information des cinq actions boursières.....	70
Tableau 5-2 : Erreur moyenne (entre les vraies (prix) et les valeurs (prix) prédites)	76
Tableau 5-3 : Erreur moyenne des cours entre les ANN et LSTM	76

Notions de base

Nous allons introduire les principes et les concepts de base, plus les définitions des termes difficiles qui ont été utilisés dans la rédaction de ce mémoire.

__ **OPCVM** : Un organisme de placement collectif en valeurs mobilières est un portefeuille dont les fonds investis sont placés en valeurs mobilières ou autres instruments financiers.

__ **LIFFE** : Le London International Financial Futures and options Exchange, ou LIFFE voire Liffe, est le marché à terme britannique, dont le principal produit est la série de contrats futurs sur l'Euribor, marché directeur des taux d'intérêt à court terme de la zone euro.

__ **OAT** : Les OAT (obligations assimilables du trésor ou valeur de trésor), sont des valeurs mobilières négociables sur les marchés financiers représentant la dette de l'État elles sont de maturité de 2 à 50 ans.

__ **Machine Learning** : « L'apprentissage automatique en français » notons ML,

__ **Intelligence artificielle** : ensemble des théories et des techniques développant des programmes informatiques complexes capables de simuler certains traits de l'intelligence humaine notent IA.

__ **Algorithme** : est une suite finie et non ambiguë d'instructions et d'opérations permettant de résoudre une classe de problèmes.

__ **Programmation classique** : ou procédurale de sorte le débutant peut la connaître à travers des langages de programmation comme Pascal, C et d'autres, Il traite les programmes comme un ensemble de données sur lesquelles agissent des procédures.

__ **LipNet** : est un réseau neuronal profond pour la reconnaissance visuelle de la parole. Il a été créé par Yannis Assael, Brendan Shillingford, Shimon Whiteson et Nando de Freitas, chercheurs de l'Université d'Oxford.

__ **Data** : un terme utilisé en français comme synonyme du mot donnée, en particulier dans le domaine informatique.

__ **BigData** : « les Méga-données ou les données massives en français » est un concept permettant de stocker un nombre indicible de données sur une base numérique.

__ **Dataset** : Un dataset en machine Learning regroupe un ensemble de données. Celles-ci dépendent d'une variable associée aux valeurs. Leur accès peut se produire de manière individuelle ou collective. Il existe différents modèles, comme le dataset d'entraînement, le dataset de test et le dataset de validation.

__ **Deep Learning** : « apprentissage profond en français » est un sous-domaine du Machine Learning, est un ensemble de méthodes du ML tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires.

__ **data Scientist** : « data science en français » combine plusieurs domaines, dont les statistiques, les méthodes scientifiques, l'intelligence artificielle et l'analyse des données, pour extraire de la valeur des données.

__ **Ressources IT** : technologies de l'information (ou IT pour Information Technology) ; les ressources IT peuvent faire référence à tout matériel, logiciel, réseau et technologie connexes que département des technologies de l'information gère et soutenant ainsi les options commerciales. Souvent, dans une entreprise il y a plus de ressources IT à gérer que d'employés.

__ **Label** : le label, est un moyen d'information du public sur les propriétés et les qualités objective d'un ouvrage, d'un environnement, d'une information, d'un bâtiment, d'une procédure et d'autre.

__ **Data Mining** : le terme Data Mining désigne l'analyse de données depuis différentes perspectives et le fait de transformer ces données en informations utiles, en établissant des relations entre les données ou en repérant des patterns.

__ **Artificiel Neural Networks** : « Réseaux de neurone artificiel » ; notons ANN.

__ **Récurrent Neural Networks** : « Réseaux de neurone récurrents » ; notons RNN.

__ **Poids** : Dans un ANN, chaque neurone est connecté aux autres neurones via des liens de connexion. Ces liens ont un poids. Le poids contient des informations sur le signal d'entrée du neurone. Les poids et le signal d'entrée sont utilisés pour obtenir une sortie. Les poids peuvent être désignés sous une forme matricielle qui est également appelée matrice de connexion. Chaque neurone est connecté à tous les autres neurones de la couche suivante via des poids de connexion.

__ **Biais** : Le biais est ajouté au réseau en ajoutant un élément d'entrée $x(b) = 1$ dans le vecteur d'entrée. Il porte également un poids désigné par $w(b)$; il joue un rôle important dans le calcul de la sortie du neurone. Il peut être positif ou négatif. Un biais positif augmente le poids net des intrants tandis que le biais négatif réduit les intrants nets.

_ **Seuil**: Une valeur seuil est utilisée dans la fonction d'activation. L'entrée nette est comparée au seuil pour obtenir la sortie. Dans ANN, la fonction d'activation est définie en fonction de la valeur de seuil et la sortie est calculée.

_ **Taux d'apprentissage** : Il est noté η , Le taux d'apprentissage varie de 0 à 1. Il est utilisé pour l'ajustement du poids pendant le processus d'apprentissage d'ANN.

_ **Long short-term memory** : « La mémoire longue à court terme » ; notons LSTM.

_ **Python** : C'est un langage de programmation interprété, multi-paradigme et multi-plateformes; Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

Introduction Générale

Depuis quelques années, l'Intelligence Artificielle connaît un regain d'intérêt sans précédent grâce à d'importantes avancées technologiques, notamment dans le domaine du Machine Learning, qui étendent les capacités des ordinateurs et accroissent leurs performances dans un grand nombre de domaines.

Parmi les domaines qui ont témoigné d'un reflet du progrès technologique nous avons choisi le marché boursier.

De nos jours la technologie joue un rôle important dans l'évolution des services boursiers. Elle vise à fournir un rendement le plus élevé possible à la fois pour l'actionnaire et l'investisseur en réduisant autant que possible les risques et les pertes. En observant les fluctuations des cours des actions, une personne peut être en mesure de prédire la direction des actions pour demain ou le lendemain, mais cette anticipation reste relative et n'apporte aucun bénéfice. Ce qui importe le plus, c'est de connaître le sort de ces actions dans le long terme.

C'est là où apparaît le rôle de l'intelligence artificielle; Les chercheurs ont créé des algorithmes d'IA plus performants et mieux adaptés pour prédire les variations des marchés.

Parmi ces algorithmes nous allons consacrer ce mémoire à *l'étude de L'Algorithme Long Short Term Memory (LSTM) pour la prédiction des cours boursiers.*

Ce mémoire a pour objectif de répondre aux deux questions suivantes:

- _ L'algorithme LSTM donne-t-il des résultats de prédictions satisfaisants?**
- _ Est-il d'une efficacité à 100% pour prédire les cours Boursiers ?**

Afin de bien cerner cette problématique, nous avons posé les sous questions suivantes :

- _ Qu'est-ce les LSTMs et quelles sont leurs fonctionnalités ?
- _ Comment prédire les cours boursiers en utilisant les LSTMs ?

Nous tenterons de répondre à ces questions à travers l'hypothèse suivante :

« La prédiction se fait après une étude des données précédente et les résultats de la prédiction peuvent ne pas être à 100% mais restent proche de la réalité ».

Notre travail est basé sur la méthode descriptive et quantitative. Nous avons adopté deux approches :

__L'approche théorique pour laquelle nous avons effectué une recherche documentaire relative au thème à travers des consultations d'ouvrages et rapports, mémoires et thèses soutenus, articles règlementaires et sites internet.

__ Pour l'approche pratique, on s'est basé essentiellement sur les deux codes du langage python "LSTM & ANN".

Cinq chapitres composent ce mémoire et structuré comme suit:

__ Le premier chapitre revêt un aspect introductif, en initialisant les notions de base relatives à la bourse.

__ Le deuxième chapitre sera consacré essentiellement sur à un des domaines de l'intelligence artificielle qui est le machine Learning; nous avons évoqué plusieurs concepts tels que les algorithmes d'apprentissage et les domaines d'applications du machine Learning.

__ Dans le troisième chapitre, on a abordé un sujet plutôt délicat avec plusieurs notions mathématiques à savoir les réseaux des neurones artificiels.

__ Le quatrième chapitre représente la partie essentielle de notre travail; il illustre les réseaux de neurones récurrents précisément la cellule LSTM de ce réseau.

__ On procédera dans le cinquième chapitre à une étude empirique de la prédiction boursière en faisant une comparaison entre deux algorithmes de prédiction "LSTM & ANN".

__ Enfin, une conclusion générale récapitulera les résultats essentiels de ce mémoire.

Chapitre 1 : Le marché boursier

*« L'argent dans une bourse entre agréablement
Mais le terme venu que nous devons le rendre
C'est lors que les douleurs commencent à nous prendre ».*

Molière

1.1 Introduction

Ce chapitre a pour objet de présenter quelques notions fondamentales du marché boursier.

1.2 Histoire rapide

❖ La création de la Bourse de Paris (1720-1830) : Au XVIII^e siècle, en France, la bourse naissante concerne avant tout le commerce des titres de la dette royale. L'allocation de fonds aux entreprises se fait alors par le biais des notaires et de banquiers.

❖ L'essor de la Bourse de Paris (1830-1870) : La révolution de 1830 engendre une crise boursière au point de mettre en difficulté de nombreux agents de change. Pendant la décennie 1850, on assiste à un doublement des volumes échangés. Une cascade de défauts va avoir lieu.

❖ L'âge d'or (1871-1913) : Le krach de 1882 est le plus violent de l'histoire boursière française. La crise financière de 1882 est circonscrite en France. La place de Paris perd donc du terrain vis-à-vis de ses rivales européennes.

❖ La bourse dans la grande transformation (1914-1944) : La France sort du conflit exsangue. Le franc-or a disparu avec la suspension de la convertibilité des billets le 5 août 1914. Un puissant secteur bancaire contrôlé par l'Etat a aussi émergé. Dans l'immédiat, après la première guerre mondiale apparaissent les chèques postaux, le crédit national, la caisse nationale et le crédit agricole. Viennent après la Banque française du commerce extérieur, les Banques populaires et le crédit coopératif.

❖ Encadrement et décloisonnement (1945-1988) : Dès les débuts de la V^eme république, le système financier va changer : les banques obtiennent davantage de latitude dans leurs choix d'investissement et l'intermédiation boursière voit ses prérogatives étendues. La relance des marchés financiers doit fournir aux entreprises des moyens de financement alternatifs aux crédits publics.

❖ La bourse dans une économie financiarisée : Les principales bourses d'Europe et d'Amérique du Nord sont devenues des sociétés privées à but lucratif et, à l'aube de l'an 2000, elles se sont elles-mêmes introduites sur les marchés qu'elles régissent. Les bourses sont alors devenues des sociétés cotées, en concurrence entre elles [2.1].

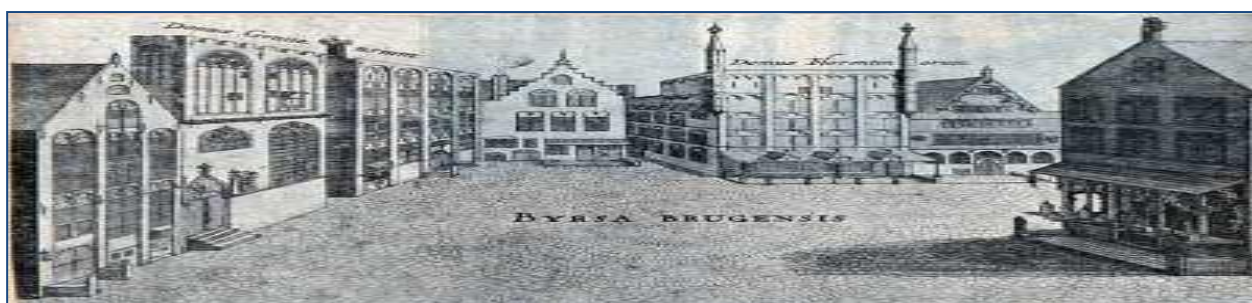


Figure 1-1 : Création de la première bourse.

1.3 Définition

La bourse est un marché financier dans lequel s'effectuent des transactions sur les valeurs mobilières et les marchandises. C'est le lieu où les actions et les obligations cotées en bourse, sont vendues et achetées par des investisseurs. Les valeurs mobilières ont une cote qui définit leur prix de vente et leur prix d'achat. Très souvent, les investisseurs ont recours à des intermédiaires pour participer à la bourse. Ces intermédiaires peuvent être des banques, des sociétés de gestion ou des conseillers financiers. L'achat ou la vente d'actions et d'obligations en Bourse nécessite, pour l'investisseur et le vendeur, de donner un ordre de bourse. Cet ordre contient des informations tels que la nature de l'opération (vente ou achat), la quantité d'actions ou d'obligations concernées par la transaction, la date limite à laquelle l'ordre doit être effectué [3.1].

La bourse appartient au marché financier. On appelle marché financier le marché des capitaux à long terme sur lequel se négocient des valeurs mobilières. Une valeur mobilière est un ensemble de titres émis par une personne morale publique ou privée, transmissibles par inscription en compte ou tradition, qui confèrent des droits identiques par catégorie et donnent accès, directement ou indirectement, à une quotité du capital de la personne morale émettrice ou à un droit de créance général sur son patrimoine.

Il existe trois types de valeurs mobilières : les actions, les obligations et les titres hybrides (valeurs à caractéristiques actions et obligations) ; Le rôle macroéconomique de la bourse s'étudie au travers de deux compartiments qui existent quelle que soit la place financière et quel que soit le marché de cotation : marché primaire et marché secondaire [1.1].

1.3.1 Marché primaire

Le marché primaire est le marché des titres neufs proposés pour la première fois en bourse, c'est-à-dire par appel public à l'épargne. Une entreprise (privée ou publique, industrielle, commerciale ou financière) ou un État propose des titres on l'appelle émetteur. En particulier, une banque ou une entreprise achète lesdits titres : on les appelle investisseurs.

Les émetteurs et les investisseurs se rencontrent sur un marché (dématérialisé) comme suit :

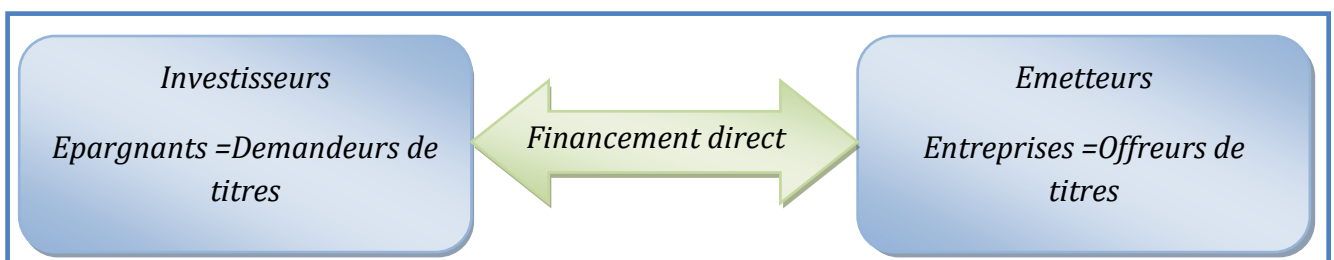


Figure 1-2 : Schéma d'un marché dématérialisé.

Lorsque l'émetteur propose pour la première fois ses titres par appel public à l'épargne, il cherche à obtenir un moyen de financement. Les investisseurs qui achètent ces titres à l'émission (sur le marché primaire) apportent donc un financement direct aux émetteurs. Le marché primaire a lieu d'être lors des opérations suivantes :

- l'introduction en bourse ;
- les privatisations;
- les augmentations de capital;
- le lancement d'un emprunt obligataire [1.1].

Immédiatement après l'émission des valeurs mobilières « neuves », on passe sur le marché secondaire.

1.3.2 Marché secondaire

Le marché secondaire est le marché des titres « d'occasion ». C'est le marché sur lequel s'échangent des titres antérieurement émis. Le passage d'un compartiment à l'autre, du marché primaire au marché secondaire, est instantané. Lorsqu'une entreprise E s'introduit en bourse, un investisseur A achète pour 100 € une action (il apporte un moyen de financement à hauteur de 100 € à l'entreprise) et peut, quelques minutes après, revendre son titre. Cette opération s'effectue sur le marché secondaire. Le rôle du marché secondaire est d'assurer la liquidité et la mobilité de l'épargne.

Globalement, une bourse se doit d'être liquide pour capter des investisseurs et des capitaux. La liquidité est une condition indispensable à la survie d'une bourse sur le plan international. La liquidité est la caractéristique d'un marché où l'on peut effectuer des opérations d'achat et de vente sans provoquer de trop fortes variations de cours. Le marché secondaire, appelé communément la bourse, répond à la loi de l'offre et de la demande [1.1].

1.4 La loi de l'offre et de la demande

La loi de l'offre et de la demande met en relation l'offre et la demande en fonction du prix et de la quantité pour déterminer un prix donné et une quantité donnée. Le terme de cette loi a été utilisé par Alfred Marshall à la fin du XIX^e siècle lorsqu'il l'a représentée graphiquement. (Figure 1-3). La courbe de demande décroît lorsque le prix augmente; la courbe d'offre suit le mouvement inverse. Le point d'intersection des deux courbes représente le prix et la quantité d'équilibre.

La bourse est le lieu de rencontre entre une offre et une demande de titres. Lorsqu'il y a confrontation des quantités offertes et des quantités demandées (donc des ordres de vente et des ordres d'achat), il y a ensuite établissement d'un cours d'équilibre (d'un prix) permettant de maximiser le nombre de titres à échanger.

Si la confrontation ne permet pas d'établir un prix d'équilibre, le prix s'ajustera à la hausse ou à la baisse pour déterminer ledit prix d'équilibre : si la quantité de l'offre (donc le nombre de titres à la vente) est trop importante par rapport à la demande (donc le nombre de titres à l'achat), le cours de bourse baisse (pour attirer des acheteurs potentiels), et inversement. Cette confrontation a lieu en continu ou à heures fixes [1.1].

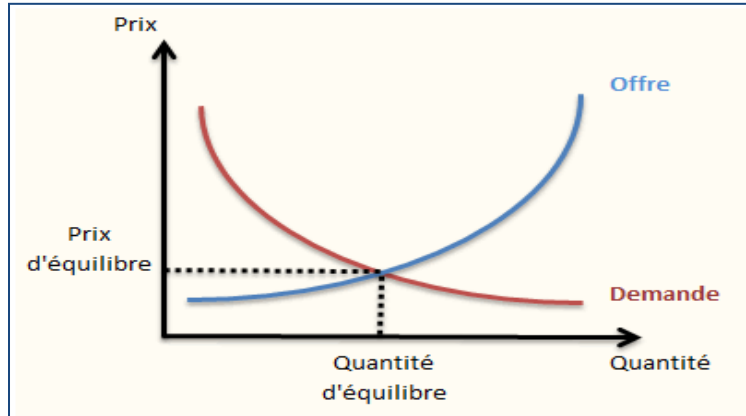


Figure 1-3 : Courbe de l'offre et la demande avec concurrence pure et parfaite.

1.5 Les titres de financement les plus courants sur le marché boursier

Les bourses se sont progressivement dématérialisées et ont migré sur des réseaux informatiques qui assurent les mêmes fonctions, sans qu'une présence physique des opérateurs en un même lieu soit nécessaire. Des opérateurs purement logiciels sont apparus dans les dernières décennies. On distingue :

- les bourses des valeurs : actions, obligations, produits dérivés (options, bons de souscription), OPCVM.
- les bourses de matières premières (métaux, hydrocarbures, céréales, diamants, etc.). L'or et l'argent ont un statut un peu particulier « compartiment » qui est dû à leur ancienne fonction monétaire.
- les bourses spécialisées dans les contrats à terme futures (matière première par exemple pétrole, produits financiers, par exemple indice boursier) comme le LIFFE.

1.5.1 Les actions

Une action est une valeur mobilière émise par une société de capitaux; l'ensemble des actions représente le capital social de cette entreprise. L'actionnaire est propriétaire de l'entreprise. Lors d'une création d'entreprise, les associés apportent des fonds pour la constitution de la société qu'on appelle le capital social. En échange, ils reçoivent des actions. Lorsque cette dernière ouvre son capital à des apporteurs de fonds propres extérieurs, elle émet des actions par appel public à l'épargne.

Les actions sont alors négociables en bourse et chaque actionnaire a des droits identiques aux autres :

- ✓ droit au dividende ;
- ✓ droit de vote ;
- ✓ droit à l'information;
- ✓ droit de mener en responsabilité des actions contre les dirigeants sociaux ;
- ✓ droit au boni de liquidation ;
- ✓ droit de souscription ou d'attribution [1.1].

Les différents types d'actions : L'action dite ordinaire est l'action de numéraire dont le montant a été libéré en espèces ou émise par incorporation de réserve au capital. Mais il existe d'autres types d'actions :

- l'action à dividende prioritaire (ADP) sans droit de vote est une action pour laquelle l'émetteur réserve un droit préférentiel sur le versement du dividende, pour un minimum égal à 7,5 % de sa valeur nominale.
- l'action préférentielle ou de priorité est qualifiée comme telle car elle offre des avantages pécuniaires (droit d'antériorité sur les bénéfices, ou attribution d'un pourcentage plus important de bénéfice, ou les deux), tout en conservant son droit de vote.
- l'action à bons de souscription d'actions est une action à laquelle sont attachés un ou plusieurs bons permettant de souscrire à de nouvelles actions à des périodes et à des conditions connues d'avance.
- l'action de préférence (créée par la loi du 24 juin 2004) peut être dotée ou non du droit de vote et peut être assortie d'un droit particulier de toute nature à titre temporaire ou permanent. Aujourd'hui, cette action remplace définitivement l'action à dividende prioritaire et le certificat d'investissement qui existent toujours sur le marché secondaire mais ne peuvent plus être émis [1.1].

1.5.2 Les obligations

Une obligation est un titre représentatif d'un emprunt émis sur les marchés financiers. L'obligataire est créancier de l'entreprise; à ce titre, il dispose de deux droits (et deux seuls droits) : percevoir les intérêts et être remboursé. En achetant des obligations, un investisseur recherche un rendement avec un risque limité.

Toutes les caractéristiques d'une obligation sont clairement stipulées dans le prospectus d'émission visé par l'AMF (Autorité des marchés financiers). Les principales caractéristiques d'une obligation sont comme suit :

- Le montant de l'émission ;
- Le prix d'émission ;
- La période de souscription « Elle est de 3 semaines environ en moyenne » ;
- La date de jouissance et de règlement ;

- Le taux d'intérêt nominal et la périodicité ;
- Le taux de rendement ;
- L'amortissement et remboursement ;
- La durée de l'émission « La durée minimale d'un emprunt obligataire est de 3 ans » ;
- Le rang de créance ;
- La garantie ;
- La notation [1.1].

Les différents types d'obligations : Parmi les principaux types d'obligations, hors obligation classique, on distingue :

- l'obligation à coupon zéro ;
- l'obligation indexée;
- l'obligation à rendement élevé;
- l'obligation à fenêtre;
- l'obligation prorogable ou échangeable;
- l'obligation foncière;
- Les obligations assimilables du trésor (OAT);
- l'obligation convertible en action;
- l'obligation à bons de souscription d'actions;
- l'obligation remboursable en actions;
- les titres subordonnés [1.1].

Valeur totale de l'obligation(VTO) : Elle correspond au prix à payer par l'acquéreur de l'obligation

$$VTO = \text{valeur cotee} + \text{coupon couru}$$

Calcul du coupon couru : Les intérêts courus sont calculés au prorata temporis (en comptant le nombre de jours exact entre la date du dernier versement d'intérêts et la date de négociation) en incluant 3 jours ouvrés (délais qui s'écoule entre la date de négociation et la date de règlement-livraison). Les jours ouvrés ne comprennent pas le vendredi et le samedi et les jours fériés.

Prix d'une obligation à une échéance : le prix d'une obligation est égale à la valeur actuelle, calculée au taux du marché obligataire en vigueur, des flux financiers qu'elle va engendrer c'est-à-dire:

$$p = \sum_k^n R_k (1 + r)^{-k}$$

Dans le cas d'un remboursement se fait en fin de période: $p = \gamma \frac{1-(1+r)^{-n}}{r} + R(1+r)^{-n}$

où :

- ✓ p : Prix de l'obligation à la date du calcul ;
- ✓ r : Taux d'intérêt du marché obligataire à la date 0 ;
- ✓ n : Durée résiduelle de l'emprunt obligataire ;
- ✓ γ (Gamma): $C \times i$ = Montant du coupon (intérêt par an et par titre) ;
- ✓ i : Taux d'intérêt nominal (ou facial) annuel ;
- ✓ C : Valeur nominale de l'obligation ;
- ✓ R : Valeur de remboursement de l'obligation ;
- ✓ R_k : Flux financier engendré par l'obligation à la date K .

1.6 Rôle et fonctionnement

1.6.1 Rôle

La bourse a un rôle primordial dans l'économie contemporaine :

- ✓ un rôle de financement où les entreprises trouvent une partie des capitaux nécessaires à leur croissance, tandis que l'Etat y finance son déficit ;
- ✓ un rôle de placement où les particuliers peuvent investir et faire fructifier leur épargne [3.2].

Et cela se présente comme suit :

__ Financement des entreprises : le marché boursier permet aux entreprises de se financer, d'investir, en mettant directement en contact l'offre et la demande par la souscription à des augmentations de capital.

__ Orientation de l'épargne : les investisseurs ayant une capacité d'épargne positive peuvent devenir les actionnaires des entreprises privées ou devenir les créanciers de ces sociétés et des collectivités publiques. Ils peuvent investir soit à l'occasion d'augmentation de capital ou d'introduction en bourse sur le marché primaire, ou s'orienter sur le marché secondaire sur lequel sont négociés les titres déjà émis.

__ Gestion du risque : la bourse permet de transférer le risque par le biais des négociations d'actions et d'obligations. Elle permet également aux investisseurs (entreprises, actionnaires, créanciers...) de se protéger du risque (risques de change, de taux, de crédit, de baisse des cours...) par l'utilisation de produits dérivés : les swaps, les futures, les contrats à terme, les options.

__ Liquidité des titres : la bourse permet la négociabilité des titres. Cette facilitation de la sortie permet d'attirer un plus grand nombre d'investisseurs et explique pourquoi les sociétés de capital-risque souhaitent que les entreprises dans lesquelles elles ont investi s'introduisent en bourse.

__ Indicateur de valeur : les cotations effectuées à la bourse permettent de mesurer la valeur attribuée par le marché à une entreprise, une matière première ou à une créance dans le temps. Les cours permettent donc de suivre l'évolution du prix de produits et l'évolution générale de la situation économique d'un pays.

__ Outil de contrôle : les sociétés cotées en bourse doivent respecter des réglementations plus strictes quant à la publication de leurs comptes. Ces entreprises sont également suivies par de multiples équipes d'analystes financiers, qui diffusent largement toutes les informations qui ont potentiellement un impact sur le cours de l'action de la société [3.3].

1.6.2 Fonctionnement

La majorité des actions sont négociées à la bourse, lieu où les acheteurs et les vendeurs se rencontrent pour convenir d'un prix. Certaines bourses sont des lieux concrets où les opérations se déroulent sur un parquet. Vous avez probablement déjà vu des images d'un parquet de la bourse, où des négociateurs lèvent nerveusement les bras en l'air, agitent la main, crient et échangent des signaux (Figure 1-4). L'autre type de bourse est virtuel (Figure 1-5); il est composé d'un réseau d'ordinateurs au sein duquel les opérations sont effectuées électroniquement.



Figure 1-4 : Ventes aux enchères.



Figure 1-5 : La bourse virtuelle.

La bourse a pour but de faciliter l'échange des valeurs mobilières entre les acheteurs et les vendeurs, réduisant ainsi les risques d'investissement. Imaginez comment il serait difficile de vendre des actions s'il fallait appeler tout le monde dans le voisinage pour trouver un acheteur. En fait, la bourse n'est rien de plus qu'un marché public super sophistiquée mettant en présence les vendeurs et les acheteurs.

La bourse de Paris : En France, les acheteurs et les vendeurs passent leurs ordres de bourse par le biais d'un intermédiaire financier (banque, société de gestion, conseiller financier...) qui transmet ceux-ci à un membre officiel de la bourse (dénommé courtier, société de bourse, entreprise d'investissement, broker).

La bourse de New York : La bourse la plus prestigieuse du monde est celle de New York. Elle est une bourse du premier type, la majorité des opérations réunissant les négociateurs face à face sur le parquet. Les ordres sont envoyés par les maisons de courtage membres de la bourse et acheminés aux délégués en bourse, qui se rendent alors en un lieu précis du parquet où les actions sont négociées. N'allez pas imaginer que la bourse de New York est encore à l'âge de pierre; les ordinateurs jouent un rôle énorme dans le processus.

La NASDAQ : Bourse virtuelle, appelée marché hors cote, dont la NASDAQ est le représentant le plus connu. Les marchés hors cote n'ont pas de lieu de négociation central ou d'agents de parquet. La négociation est effectuée par l'entremise d'un réseau d'ordinateurs et de télécommunication reliant les courtiers [3.4].

1.7 Les ETF

« *Exchange-Traded Fund* » ou « *Fonds négocié en bourse* »

Les ETF sont déjà utilisés depuis les années 1990 par les investisseurs institutionnels, tels que les banques privées, les caisses de pension et les organismes publics. Depuis quelques années, ils suscitent un engouement croissant auprès des investisseurs privés. Aujourd'hui, le capital investi dans les ETF au niveau mondial est de 7,86 trillions de dollars [2.2].

Qu'est-ce qu'un ETF ? : Un EFT est un fond indiciel ou panier de valeurs se négociant en bourse. Ce fond va donc suivre l'évolution d'un indice boursier (comme le Cac 40, Euro Stock 50 ou encore le Nasdaq), à la hausse comme à la baisse [3.5].

À quoi visent les ETF : Un ETF vise à reproduire l'évolution de la valeur (la performance) d'un indice¹ donné et à dégager le même rendement (déduction faite des frais liés à l'ETF) que cet indice. Un ETF EURO STOXX 50 (Figure 1-6).

Par exemple, il vise à reproduire la performance de l'indice boursier européen. Si l'indice progresse, l'ETF suit également cette tendance, et inversement si la performance baisse. Comme leur nom l'indique, les ETF sont négociés en bourse, à l'instar des actions, à un cours donné. Ils peuvent être achetés et vendus à leur cours actuel en quelques clics sur des portails en ligne. Et si vos objectifs de placement évoluent, vous pouvez à tout

¹ Un indice s'apparente à un panier de titres qui doit représenter l'évolution d'un certain marché ou segment de marché. L'Euro Stoxx 50, par exemple, regroupe les 50 principales entreprises de la zone euro. Tel un baromètre, il reflète l'évolution des positions incluses au sein de l'indice au fil du temps et sert ainsi d'étalon de comparaison pour les investisseurs et les gestionnaires de fonds.

moment adapter votre investissement en ETF (un avantage décisif par rapport aux fonds de placement moins flexibles) [2.2].

Quels sont les avantages des ETF ? : Les ETF présentent divers avantages, à commencer par la possibilité d'investir sur l'ensemble d'un marché au lieu de choisir des actions distinctes, permettant une réelle diversification. En outre, les frais y sont relativement peu élevés, et aucun minimum de souscription n'est requis. En outre, ils s'échangent très facilement, tout au long de la journée.

Comment investir dans des ETF ? : Investir dans des ETF peut se faire soit sur le marché primaire, en contactant un intermédiaire financier agréé, soit sur le marché secondaire, c'est-à-dire en bourse, de la même manière que des actions. A noter qu'il est également possible d'investir dans un ETF par le biais d'une assurance-vie sous forme d'unité de compte.

Quels sont les risques des ETF ? : Il existe plusieurs risques liés aux investissements dans les ETF. Le risque le plus connu et le plus évident est la variation de l'indice suivi. Lorsque l'indice baisse, le portefeuille diminue dans les mêmes proportions. La baisse sera d'autant plus forte pour ceux qui investissent dans un ETF à effet de levier. Un autre risque important à prendre en compte est le risque de change lorsque l'ETF est coté en devise étrangère. Précisons enfin que certains ETF sont très peu liquides, car peu connus. Il sera alors difficile de trouver des acheteurs potentiels. En sens inverse, un ETF très liquide n'est pas un gage de sécurité pour l'investisseur, car la possibilité d'effectuer de très nombreux mouvements au cours d'une même journée peut engendrer des frais élevés [3.5].

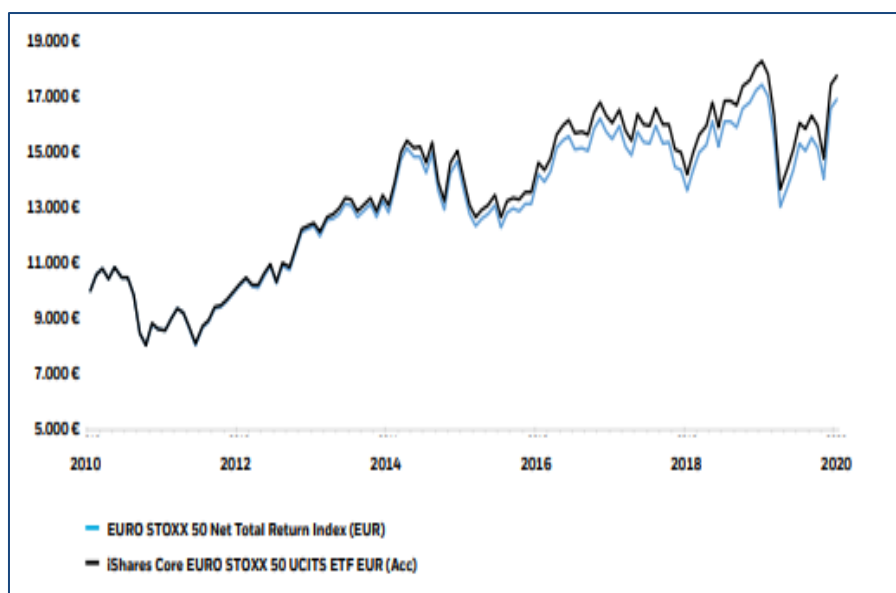


Figure 1-6 : Exemple de la reproduction de l'évolution de la valeur de L'EURO STOXX 50.

1.8 La prédiction boursière

Il est impossible de prédire à coup sûr le cours de bourse d'une action qui évolue en fonction de l'offre et de la demande et dépend de nombreux éléments. Plusieurs stratégies permettent néanmoins de spéculer sur sa possible évolution. Le prix d'un titre financier évolue en fonction de l'offre et de la demande sur les marchés. Ces deux facteurs fluctuent au gré de nombreux événements pouvant survenir de manière inattendue. Ils peuvent être propres à l'entreprise ou liés à son environnement.

- Dans le premier cas, le cours d'une action peut par exemple s'envoler après l'annonce d'une acquisition surprise, ou chuter si elle lance un avertissement sur résultats (profit warning).
- Dans le second cas, la valeur d'une entreprise cotée peut augmenter lorsque l'un de ses concurrents révèle un recul de ses parts de marché à son profit, ou baisser si un pays adopte une réglementation défavorable à ses activités (taxation, interdiction de produits...).

Les investisseurs n'ont, certes, pas de boule de cristal pour voir l'avenir. Mais plusieurs indicateurs peuvent permettre à un instant T de se faire une opinion sur le cours qu'une action est susceptible d'atteindre dans le futur. Si la valeur est suivie par des analystes financiers, la première étape consiste à regarder leurs objectifs de cours pour le titre. La moyenne ou la médiane de ces prévisions est appelée le consensus. Il est probablement consultable sur le site Internet de votre courtier. Les objectifs de cours des analystes sont souvent des projections à un an et se basent sur des données objectives (résultats financiers des entreprises, croissance du marché, ratios de valorisation boursière...) et subjectives (appréciation de la qualité des dirigeants, perception de l'environnement concurrentiel, prédictions de bénéfices futurs...). [2.3].

La prédiction boursière avec l'intelligence artificielle: Analyser parfaitement les mouvements boursiers pour anticiper l'avenir semblait être une tâche impossible, jusqu'à ce que l'entrepreneur Jeff Glickman affirme avoir mis au point une intelligence artificielle capable de prévoir les courbes.

- dans les années 2000, qu'il a l'idée de s'appuyer sur un modèle mathématique s'appliquant aux particules pour développer une IA capable de prédire le cours de la Bourse. «Mais peu importe ce que j'essayais, rien ne marchait sur les marchés boursiers», confie Glickman à Fast Company.
- En 2004, l'Américain réalise qu'il a besoin de travailler avec un autre genre de logiciel: une intelligence artificielle en mesure de prouver des théorèmes et de se reprogrammer elle-même, afin de développer de nouveaux modes de traitement des données financières. S'il faudra encore dix ans à Glickman pour affiner son concept, l'homme a senti très tôt que son idée tenait la route et que le défi n'était plus hors de portée.
- En 2015, il allume enfin son superordinateur, alimenté par 400 serveurs.

Pendant vingt-quatre heures, la machine réalise des calculs sous la surveillance de Glickman et de son associé. À la fin de la journée, témoigne-t-il, «on a regardé ce qu'on avait obtenu, on s'est gratté la tête et on a réalisé: "Oh mon Dieu, nous n'avons aucune idée de ce qu'elle vient de faire."»; Une année supplémentaire est nécessaire pour décrypter les huit premières heures de travail de l'intelligence artificielle et la logique qu'elle a développée. Mais le jeu en vaut la chandelle, les deux ingénieurs constatent qu'elle apprend et qu'elle a déjà compris comment fonctionnait le marché boursier; tout ce qui lui manque, c'est de l'entraînement [3.6].

1.9 Conclusion

La bourse a un rôle majeur du point de vue économique et financier. Elle joue le double rôle d'être à la fois un lieu de financement pour les entreprises (émission d'actions ou d'obligations) mais également un lieu de placement (investisseurs).

Chapitre 2 : Machine Learning

« Une année de travail sur l'intelligence artificielle

Est suffisante pour vous faire

Croire en Dieu ».

Alan Jay Perlis

2.1 Introduction

Le Machine Learning est utilisé par chacun de nous, des centaines de fois par jour sans se rendre compte. Chaque fois qu'on fait une recherche sur Google, une des raisons pour laquelle on tombe sur des résultats plutôt adéquats est que derrière cette action, c'est un algorithme du ML, qui a appris comment trouver les résultats les plus pertinents parmi des milliards de résultats possibles. Cet apprentissage permet d'avoir un système qui s'optimise en fonction de l'environnement, les expériences et les résultats observés. Faire tout ceci serait impossible avec la programmation classique car il faudrait coder des milliards de cas possibles. Cependant, avec le ML on donne à un algorithme la capacité d'apprendre et tout devient beaucoup plus facile.

Ce chapitre sera notre première porte d'entrée dans le domaine de l'apprentissage automatique, nous aurons une idée générale de la façon dont la machine peut apprendre pour donner des résultats plutôt fiables.

2.2 Historique

Le ML a émergé dans la seconde moitié du XX^{ème} siècle du domaine de l'IA.

- ❖ les premières traces de l'IA remontent à 1950 dans un article d'Alan Turing intitulé "Computing Machinery and Intelligence" dans lequel le mathématicien explore le problème de définir si une machine est consciente ou non.

- ❖ en 1957, un autre informaticien américain, Frank Rosenblatt, crée le « perceptron » c'est-à-dire un algorithme qui constitue le premier réseau de neurones artificiels. Nous parlerons de ça en détail dans le chapitre suivant.

- ❖ En 1959, c'est l'informaticien américain Arthur Samuel qui utilise pour la première fois le terme « Machine Learning », pour son programme créé en 1952.

- ❖ On assiste alors à un coup d'arrêt significatif pour l'IA et le ML ; Tant et si bien qu'on parle d'« hiver de l'IA » à partir de 1974. Et si la technologie connaît un rebond au début des années 1980, elle doit affronter une nouvelle crise de 1987 à 1993. Une période considérée comme le « deuxième hiver de l'IA ».

- ❖ le début des années 1990 a marqué le renouveau du Machine Learning.

- ❖ Au début des années 2000, la recherche en intelligence artificielle change de paradigme et opte pour le Machine Learning. Jusqu'alors, pour enseigner quelque chose à la machine, il fallait encoder manuellement chaque règle à suivre.

- ❖ En 2012, un réseau neuronal développé par Google parvient à reconnaître des visages humains ainsi que des chats dans des vidéos YouTube.

- ❖ En 2016, un système d'intelligence artificielle à base d'apprentissage automatique nommé LipNet parvient à lire sur les lèvres avec un grand taux de succès [2.4].

Tout cela pour avoir à la fin le Machine Learning de nos jours.

2.3 Définition

Le Machine Learning est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques, pour donner aux ordinateurs la capacité d'apprendre à partir des données, c'est à-dire d'améliorer leurs performances à résoudre des tâches sans être explicite. Au cours des dernières années, le ML est devenu l'une des branches les plus importantes et les plus précieuses de l'intelligence artificielle [1.2]. Il est un domaine scientifique et plus particulièrement une sous-catégorie de l'IA, c'est une branche évolutive des algorithmes de calcul conçus pour imiter l'intelligence humaine en apprenant de l'environnement environnant. [1.3].

Tout ce qui peut être stocké numériquement peut servir de data pour le ML ; En décelant les patterns dans ces données, les algorithmes apprennent et améliorent leurs performances dans l'exécution d'une tâche spécifique. Il englobe un large éventail d'algorithmes et d'outils de modélisation utilisés pour une vaste gamme de tâches de traitement de données, qui sont entrés dans la plupart des disciplines scientifiques ces dernières années [1.4].

Le ML révèle tout son potentiel dans les situations où des insights (tendances) doivent être repérés à partir de vastes ensembles de données diverses et variées, appelés le BigData. Ces deux derniers sont explicitement liés étant donné que pour apprendre et se développer, les ordinateurs ont besoin de flux de données à analyser sur lesquelles s'entraîner. De ce fait, le Machine Learning, issu par essence du BigData, a précisément besoin de ce dernier pour fonctionner ; li est aussi réellement la science idéale pour tirer profit du BigData et de ses opportunités. Cette technologie est en effet capable d'extraire les données de valeur parmi d'immenses sources d'informations complexes, et ce sans avoir à faire appel aux humains. Entièrement dirigé par les données.

Le Machine Learning convient donc parfaitement à la complexité du BigData dont il est réellement indissociable.

2.4 Les notions clés du Machine Learning

1. Le Dataset: En Machine Learning, tout démarre d'un Dataset qui contient nos données. Dans l'apprentissage supervisé, le Dataset contient les questions (x) et les réponses (y) au problème que la machine doit résoudre.

2. Le modèle et ses paramètres: A partir de ce Dataset, on crée un modèle, qui n'est autre qu'une fonction mathématique. Les coefficients de cette fonction sont les paramètres du modèle.

3. La fonction coût: Lorsqu'on teste notre modèle sur le Dataset, celui-ci nous donne des erreurs. L'ensemble de ces erreurs, c'est ce qu'on appelle la fonction coût.

4. L'algorithme d'apprentissage: L'idée centrale du Machine Learning, c'est de laisser la machine trouver quels sont les paramètres de notre modèle qui minimisent la fonction coût (nous parleront de cela dans le prochain chapitre); Pour minimiser cette fonction coût, on utilise un algorithme d'apprentissage. Il en existe beaucoup. Un des plus connus est l'algorithme de la descente du gradient [2.15].

2.4.1 Machine Learning « Step by Step »

Voici les sept étapes clés qui résultent de ces expériences pour construire une bonne prédiction d'un modèle de Machine Learning:

- Step1 : Rassembler des données " *Gathering DATA* " ;
- Step2 : Préparation des données (80%_20%) " *DATA Preparation* " ;
- Step3 : Choisir le modèle " *choose model* " ;
- Step4 : Formation & entraînement " *training* " ;
- Step5 : Evaluation " *evaluation* " ;
- Step6 : Réglage des hyper-paramètres " *hyperparameter tuning* " ;
- Step7 : Prédiction " *prediction* " .

2.5 BigData & Deep Learning

2.5.1 Machine Learning et BigData deux notions indissociables

Le ML et le Deep Learning font partis de l'intelligence artificielle, le Deep Learning est une sous-catégorie du ML.

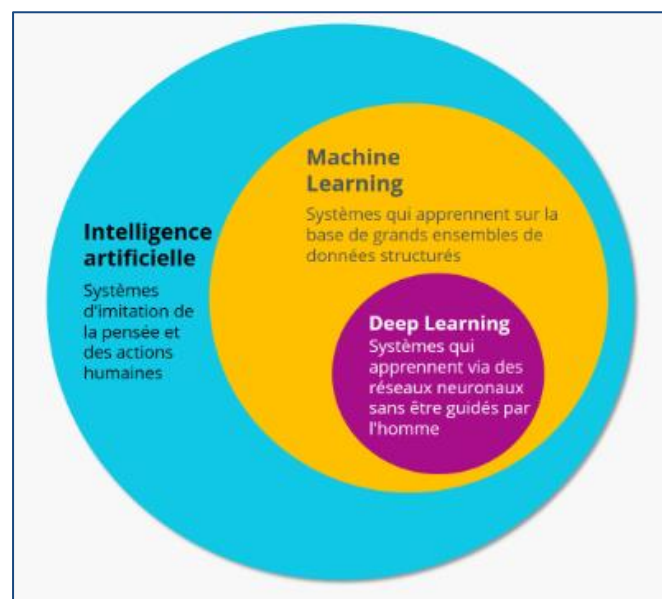


Figure 2-1 : Classement des concepts.

Le ML libère en permanence sa puissance dans un large éventail d'applications. Il a été poussé au premier plan ces dernières années en partie grâce à l'avènement du BigData. Les algorithmes du Machine Learning n'ont jamais été aussi prometteurs lorsqu'ils sont mis au défi par le BigData. Les méga-données permettent aux algorithmes de ML de découvrir des modèles plus fins et de faire des prédictions plus opportunes et plus précises que jamais [2.5].

Sans le BigData, l'IA et le ML ne seraient rien. En effet, les données sont l'instrument indispensable permettant à l'IA de comprendre et d'apprendre la manière dont l'intelligence humaine analyse les situations.

C'est donc le BigData qui permet l'automatisation des traitements de données en permettant l'accélération de la courbe d'apprentissage des ordinateurs s'appuyant sur le ML.

2.5.2 Machine Learning vs Deep Learning

Contrairement au ML, le Deep Learning n'a pas besoin de données structurées. Le système fonctionne à partir de plusieurs couches de réseaux neuronaux (on parlera des réseaux neuronaux dans le chapitre suivant), qui combinent différents algorithmes en s'inspirant du cerveau humain. Ainsi, le système est capable de travailler à partir de données non structurées. Avec le Deep Learning, le système identifie lui-même les caractéristiques discriminantes des données, sans avoir besoin d'une catégorisation préalable. Le système n'a pas besoin d'être entraîné par un développeur. Il évalue lui-même le besoin de modifier le classement ou de créer des catégories inédites en fonction des nouvelles données.

Tandis que le ML fonctionne à partir d'une base de données contrôlable, le Deep Learning a besoin d'un volume de données bien plus considérable. Le système doit disposer de plus de 100 millions d'entrées pour donner des résultats fiables.

Par ailleurs, la technologie nécessaire pour le Deep Learning est plus sophistiquée. Elle exige plus de ressources IT et s'avère nettement plus coûteuse que le ML, elle n'est donc pas intéressante, du moins à l'heure actuelle, pour une utilisation de masse par les entreprises [3.7].

2.6 Les principaux algorithmes de Machine Learning

Les types d'algorithmes d'apprentissage courants pour le Machine Learning sont comme suit :

- Supervised learning "Apprentissage supervisé" ;
- Unsupervised learning "Apprentissage non supervisé" ;
- Reinforcement learning "Apprentissage par renforcement" ;
- Semi-supervised "Semi-supervisé " [2.6].

2.6.1 Apprentissage supervisé

➤ *Définition* : Des algorithmes d'apprentissage supervisé sont utilisés lorsque la sortie est classée ou étiquetée. Ces algorithmes apprennent des données passées qui sont entrées, appelées données d'entraînement, exécutent leur analyse et utilisent cette analyse pour prédire les événements futurs de toute nouvelle donnée dans les classifications connues. La prédiction précise des données de test nécessite des données volumineuses pour avoir une compréhension suffisante des modèles. L'algorithme peut être entraîné davantage en comparant les résultats d'apprentissage aux résultats réels et en utilisant les erreurs pour la modification des algorithmes.

➤ *Exemple réel:*

_ Classification d'image : L'algorithme est tiré de l'alimentation avec des données d'image étiquetées. Un algorithme est formé et il est prévu que dans le cas de la nouvelle image, l'algorithme la classe correctement.

_ Prédiction du marché : Il est également appelé régression. Les données historiques du marché des entreprises sont transmises à l'ordinateur. Avec l'algorithme d'analyse et de régression, un nouveau prix pour l'avenir est prévu en fonction des variables.

➤ Régression linéaire : Sans aucun doute possible, les algorithmes de régression linéaire sont les plus utilisés par les équipes de data science. Il s'agit d'effectuer des corrélations simples entre deux variables dans un jeu de données. Un ensemble d'entrées et de sorties correspondantes sont examinés et quantifiés pour montrer une relation. La popularité de la régression linéaire s'explique par sa simplicité. L'algorithme est facilement explicable, relativement transparent et il y a peu de paramètres à configurer. Bien connu dans la pratique des statistiques, ce type d'algorithmes est souvent utilisé pour prévoir des ventes ou des risques.

➤ Machine à vecteurs de support (SVM) : SVM (Séparateurs à vastes marges) sont des algorithmes qui séparent les données en classes. Pendant l'entraînement, un SVM trouve une ligne qui sépare les données d'un jeu en classes spécifiques et maximise les marges de chaque classe. Après avoir appris les lignes de classification, le modèle peut ensuite les appliquer aux nouvelles données. Les machines à vecteurs de support sont très utilisées dans la finance. Elles offrent une grande précision sur les données actuelles et futures. Les modèles associés peuvent servir à comparer virtuellement les performances financières relatives, la valeur et les retours sur investissement.

➤ Arbre de décision : Un algorithme d'arbre de décision représente graphiquement les données en branches pour montrer les résultats possibles de diverses actions. Il classe et prédit les variables de réponse en fonction des décisions passées. Cette méthode visuelle a fait ses preuves.

Cependant, les arbres de décisions deviennent difficiles à lire quand ils sont associés à de gros volumes de données et à des variables complexes. C'est pourquoi ils sont utilisés pour les décisions à faibles enjeux, comme l'anticipation des variations de taux d'emprunt ou les réactions du marché si une entreprise modifie un élément important d'un de ses produits.

2.6.2 Apprentissage non supervisé

➤ Définition : Les algorithmes dits non supervisés ne sont pas entraînés par le Data scientist. Ils dépendent de méthodes d'apprentissage approfondi pour identifier des patterns en passant au peigne fin des ensembles de données d'entraînement non étiquetées, puis en observant les corrélations. Les modèles entraînés avec cette méthode

ne sont pas dirigés pour trouver un résultat ou identifier des données en particulier. Ils sont utilisés lorsque nous ignorons les résultats finaux et que la classification ou les résultats étiquetés ne sont pas à notre disposition.

Ces algorithmes étudient et génèrent une fonction pour décrire des modèles complètement cachés et non étiquetés. Par conséquent, il n'y a pas de sortie correcte, mais il étudie les données pour donner des structures inconnues dans les données non étiquetées.

➤ Exemple réel:

_ Données de haute dimension : Les données de haute dimension ne sont normalement pas faciles à utiliser. Grâce à un apprentissage non supervisé, la visualisation de données de grande dimension devient possible

_ Modèles génératifs : Une fois que votre algorithme analyse et fournit la distribution de probabilité de l'entrée, il peut être utilisé pour générer de nouvelles données. Cela s'avère très utile en cas de données manquantes.

➤ Les algorithmes APriori : L'algorithme APriori est très utilisé par les équipes commerciales qui cherchent à savoir quels produits un client va-t-il possiblement acquérir avec un autre. Cet algorithme d'exploration de données cherche les affinités entre deux éléments d'un jeu de données afin d'identifier s'il y a une corrélation négative ou positive entre eux.

Les éditeurs de moteur de recherche les utilisent pour prédire la prochaine requête d'un internaute, tandis que Netflix l'utilise comme un outil de recommandation du prochain contenu à regarder.

➤ La répartition en K-moyennes (K-means) : L'algorithme K-means s'appuie sur une méthode itérative pour trier des points de données en groupes basés sur des caractéristiques similaires. Ce type d'algorithme est aussi bien utilisé par les éditeurs de moteur de recherche pour proposer des résultats pertinents ou par des entreprises qui veulent classifier les comportements des utilisateurs. Cette technique s'avère également efficace dans le contexte d'analyse de performances informatiques.

2.6.3 Apprentissage par renforcement

➤ Définition : Les algorithmes d'apprentissage par renforcement sont basés sur des systèmes de récompenses et de punitions. L'algorithme se voit assigner un objectif et cherche à s'en rapprocher pour obtenir une récompense maximale. Il se base sur des informations limitées et apprend de ses actions précédentes. Ces algorithmes peuvent dépendre d'un modèle ; Ils doivent alors suivre des étapes prédéfinies et le nombre d'erreurs et d'essais est limité. D'autres ne se reposent pas sur un modèle et interprètent à chaque nouvel essai [2.7].

➤ Q-Learning : Les algorithmes de Q-Learning cherchent à trouver la meilleure méthode (une politique optimale) pour atteindre un objectif défini en cherchant à

obtenir un maximum de récompenses. Ils tentent le plus grand nombre d'actions possibles par état du système sans avoir de connaissance initiale de l'environnement. Un algorithme de ce type peut être construit pour obtenir rapidement des récompenses ou pour atteindre un objectif majeur. Le Q-Learning est souvent associé à des modèles de Deep Learning dans le cadre de projets de recherche.

➤ Algorithme basé sur un modèle (model-based) : À l'inverse du Q-Learning, les algorithmes basés sur un modèle ont une liberté limitée pour créer des états et des actions. Cela leur apporte néanmoins une efficacité statistique supérieure. Ils sont formés avec des données spécifiques et des actions de base en provenance de l'environnement via un entraînement supervisé. Cela permet en principe d'accélérer l'apprentissage. Il est possible de combiner cette méthode avec du Q Learning.

2.6.4 Apprentissage Semi-supervisé

➤ Définition : Les méthodes d'apprentissage semi-supervisé combinent données étiquetées et non étiquetées. Les algorithmes de ce type se nourrissent de certaines informations grâce à des catégories labélisées, des suggestions et des exemples. Ensuite, ils créent leurs propres labels en explorant les données par eux-mêmes, en suivant un schéma rudimentaire ou les indications de Data Scientists.

➤ Réseaux antagonistes génératifs : Les réseaux antagonistes génératifs (generative adversarial networks ou GAN en anglais) sont des modèles qui imitent la distribution de données. Deux réseaux sont placés en compétition afin de déterminer la meilleure solution à un problème. Un des réseaux neuronaux, appelé générateur, se nourrit des données d'entrée pour générer une sortie passable, tandis que le second, le discriminateur, s'appuie sur la sortie du premier pour y repérer les défauts et l'améliorer. Ce processus est répété autant de fois que nécessaire pour trouver une réponse idéale à un problème.

➤ Classificateur bayésien naïf : Le classificateur bayésien naïf (Naive Bayes) s'appuie sur le théorème de Bayes fondé sur les probabilités conditionnelles. Cet algorithme est utilisé par les chercheurs pour reconnaître des classes d'objets sur jeux de données étiquetés. Ensuite, l'algorithme est entraîné sur des données non étiquetées. Une fois ce cycle terminé, les chercheurs associent les étiquettes et relancent l'entraînement.

Cette technique est particulièrement utilisée dans le cadre du traitement du langage naturel ou pour labéliser des jeux de données sans faire appel à des services comme Amazon.

2.6.5 Récapitulation

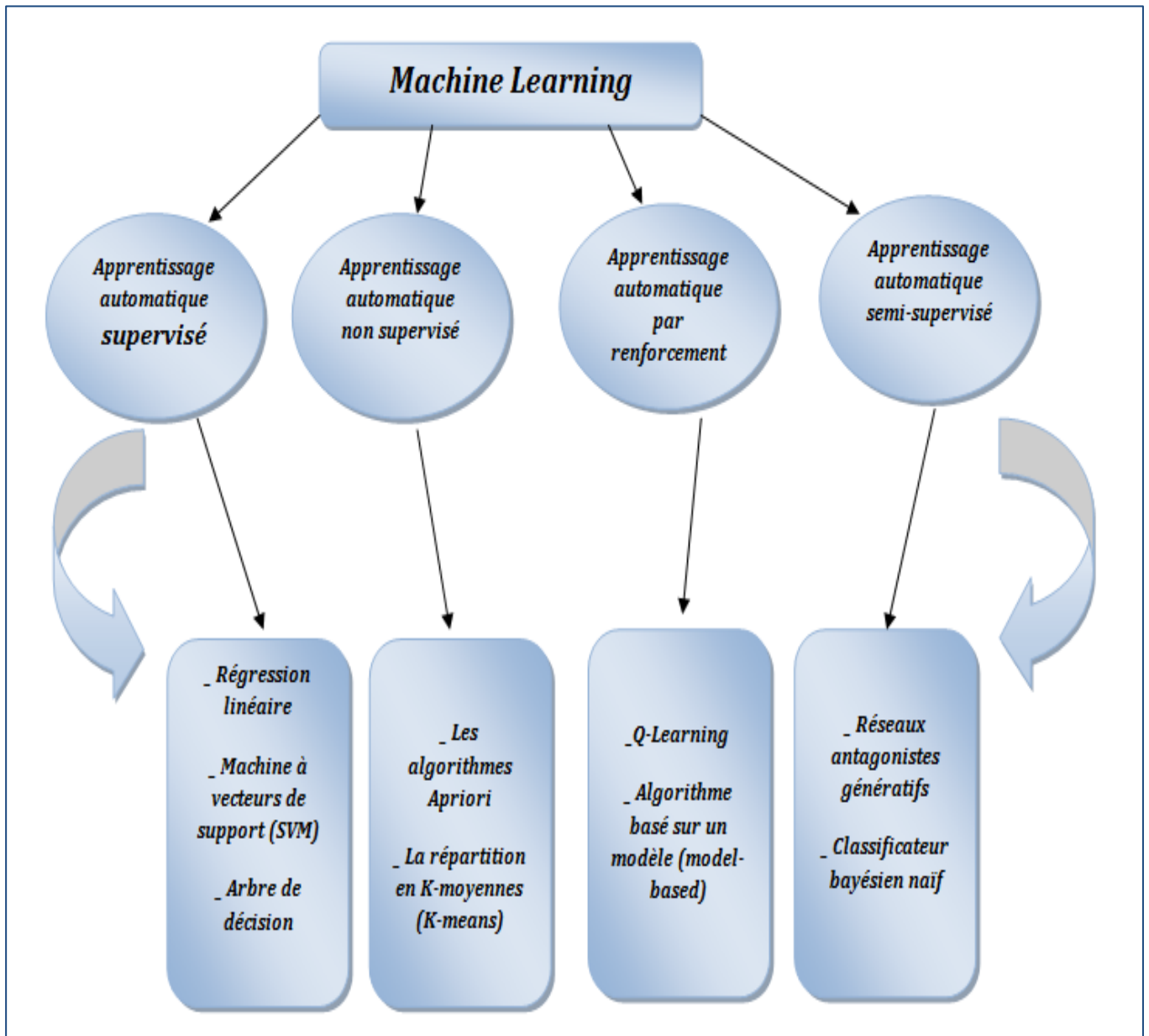


Figure 2-2 : Les principaux algorithmes du Machine Learning.

2.7 Modèle de Machine Learning

2.7.1 Définition

Un modèle de ML est un fichier qui a été entraîné à partir d'une base de connaissances en vue d'automatiser des tâches, par exemple reconnaître une émotion au regard d'une expression sur un visage, traduire un texte, proposer des produits en fonction d'un profil d'appétence... Une fois entraîné, le modèle doit être capable de générer des résultats à partir de données (textes, photos) qu'il n'a encore jamais traité [2.8].

2.7.2 Développement d'un modèle de Machine Learning

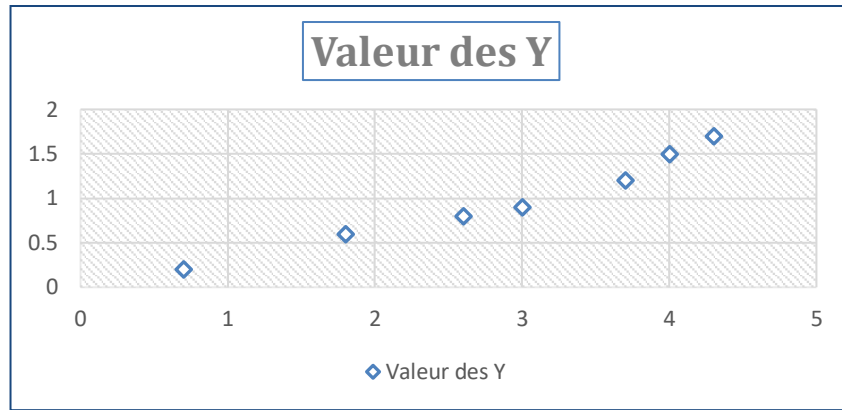
Le développement d'un modèle de Machine Learning repose sur quatre étapes principales. En règle générale, c'est un Data Scientist qui gère et supervise ce procédé.

1. La première étape consiste à sélectionner et à préparer un ensemble de données d'entraînement. Ces données seront utilisées pour nourrir le modèle de Machine Learning pour apprendre à résoudre le problème pour lequel il est conçu.
2. La deuxième étape consiste à sélectionner un algorithme à exécuter sur l'ensemble de données d'entraînement. Le type d'algorithme à utiliser dépend du type et du volume de données d'entraînement et du type de problème à résoudre.
3. La troisième étape est l'entraînement de l'algorithme. Il s'agit d'un processus itératif. Des variables sont exécutées à travers l'algorithme, et les résultats sont comparés avec ceux qu'il aurait dû produire. Les "poids" et les "biais" peuvent ensuite être ajustés pour accroître la précision du résultat.
4. La quatrième et dernière étape est l'utilisation et l'amélioration du modèle. On utilise le modèle sur de nouvelles données, dont la provenance dépend du problème à résoudre.

2.8 Fonctionnement et domaine d'application

2.8.1 Pourquoi le Machine Learning est utilisé

Pour comprendre au mieux ce qu'est le Machine Learning et comment cela fonctionne, il faut commencer par comprendre pourquoi il est utilisé. Nous, les êtres humains, sommes quotidiennement confrontés à des problèmes que nous cherchons à résoudre. Par exemple, représenter des données et comment les utiliser pour développer un modèle « par exemple une fonction $F(X)=AX+B$ ».

Figure 2-3 : Fonction $F(X)=AX+B$

Pour nous aider dans nos recherches, nous avons inventé l'ordinateur, qui permet de résoudre la fonction, pour trouver les paramètres A et B qui donne le meilleur modèle possible et tout cela en quelques minutes, des calculs qui nous prennent beaucoup de temps à être effectués. Mais il faut savoir qu'un ordinateur ne sait en réalité faire qu'une chose; résoudre les calculs qu'on lui donne.

Pour cela, on programme dans la machine un algorithme d'optimisation qui va venir tester différentes valeurs de A et B, jusqu'à obtenir la combinaison qui minimise la distance entre le modèle et les points. Le ML développe un modèle, en se servant d'un algorithme d'optimisation pour minimiser les erreurs entre le modèle et nos données.

2.8.2 Fonctionnement

Le Machine Learning consiste à écrire un programme qui au début ne sait rien faire, mais qui va apprendre à faire quelque chose avec le temps et l'expérience. La machine récupère des quantités gigantesques de DATA, qu'elle réutilise (analyse) pour s'adapter à de nouvelles situations (création des modèles) et même pour les anticiper (anticipation, prédiction). L'idée est que l'algorithme construise une « représentation interne » tout seul, une sorte de conscience de la situation devant laquelle il se retrouve, afin de pouvoir effectuer la tâche qui lui est demandée (prédiction, identification, etc.).

Pour résumer le ML se compose de 2 phases:

1) *Training*: La première phase est la conception du système qu'on appelle aussi phase d'apprentissage ou d'entraînement. C'est l'estimation d'un modèle à partir de l'analyse des données. Cela comprend une estimation d'une densité de probabilité ou la résolution d'une tâche pratique.

2) *Prediction*: La seconde phase est une mise en production. Il est possible que des systèmes continuent leur apprentissage même en étant déjà en production. Après la détermination du modèle, on teste la seconde partie de données utile pour la réalisation de la tâche désirée.

2.8.3 *Domaine d'application*

Le commerce de détail, les médias, la médecine, la finance, la science, l'industrie, la sécurité et le gouvernement dépendent des prédictions produites par des techniques telles que l'apprentissage automatique [2.9].

Aussi le ML peut résoudre des problèmes comme :

- Un problème qui nécessite de nombreuses et longues listes de règles pour trouver la solution. Dans ce cas, les techniques du ML peuvent simplifier votre code et améliorer les performances.
- Problèmes très complexes pour lesquels il n'y a pas de solution avec un approche.
- Environnements instables : les logiciels d'apprentissage automatique peuvent s'adapter aux nouvelles données.

2.8.4 *Dans le monde de la finance*

Récemment, on constate la présence de nombreuses études analysant les comportements et prédictions du prix des actions et les variations d'indices, impliquant des algorithmes de Machine Learning. De nos jours, les traders utilisent des systèmes de commerce intelligent pour mieux prédire l'évolution des prix face à différentes situations et conditions, ce qui facilite la prise de décision des investisseurs.

Normalement, un bon trader prédit le prix des actions et achète une action si son prix va augmenter dans un avenir proche, et revend une action si son prix s'apprête à diminuer. Cependant, la volatilité expliquée ci-dessus rend la tâche des traders bien difficile, car il est impossible de prédire avec 100% d'exactitude l'évolution du prix d'une action. C'est dans ce cadre-là que les investisseurs et gestionnaires de portefeuille ont commencé à se pencher vers le Machine Learning. Bien qu'il soit difficile de remplacer les compétences acquises par un trader expérimenté, un algorithme de prédiction précis peut également engendrer des bénéfices pour des entreprises d'investissements, mettant en évidence une relation entre la précision de l'algorithme et les profits de son utilisation [2.10].

2.8.5 *Analyse prédictive*

Réaliser une analyse prédictive consiste à exploiter les données issues du Big Data et traitées par des algorithmes statistiques et des techniques de ML, afin de prédire des probabilités en se basant sur le passé. Les analyses prédictives sont réalisées à partir de plusieurs disciplines et technologies telles que le Data Mining, les analyses statistiques et bien sûr le ML, afin de permettre aux entreprises de prévoir les tendances et résultats financiers de demain.

Ces analyses prédictives sont le but ultime du ML. Elles permettent d'extraire des insights exploitables à partir de larges ensembles de données, afin d'offrir aux

entreprises la possibilité de prendre de meilleures décisions stratégiques, en accord avec leurs objectifs.

L'IA et le ML constituent sans aucun doute le niveau supérieur de l'analyse de données, les systèmes informatiques étant désormais capables d'apprendre en permanence des données captées par les entreprises afin de prédire intelligemment les besoins des consommateurs, les tendances futures de tel ou tel marché et bien plus encore.

2.9 Conclusion

Le Machine Learning est un outil très puissant qui permet d'effectuer des multiples actions comme classifier des données, faire apprendre à un programme à partir d'expérimentations ou encore de créer un programme évolutionnaire qui s'améliore sans cesse et faire des prédictions précises en fonction des modèles observés.

Chapitre 3 : Les réseaux de neurones artificiels

*« Il y a des circonstances de la vie où l'homme ressemble à un ordinateur
Tout lisse à l'extérieur mais clignotant des neurones avec frénésie ».*

DANIEL PENNAC

La plus grande partie de ce chapitre a adopté la référence [1.10].

3.1 Introduction

Les réseaux de neurones sont l'un des plus beaux paradigmes de programmation jamais inventés. Dans l'approche conventionnelle de la programmation, nous disons à l'ordinateur ce qu'il doit faire, en divisant les gros problèmes en de nombreuses petites tâches définies avec précision que l'ordinateur peut facilement effectuer.

En revanche, dans un réseau de neurones, nous ne disons pas à l'ordinateur comment résoudre notre problème. Au lieu de cela, il apprend à partir des données d'observation, trouvant sa propre solution au problème posé.

Les réseaux de neurones artificiels sont des modèles électroniques relativement rudimentaires basés sur la structure neuronale du cerveau.

Dans ce chapitre, nous traiterons le concept d'un réseau de neurone avec un point de vue mathématique et théorique.

3.2 Historique

❖ 1890 : W. James, célèbre psychologue américain introduit le concept de mémoire associative, et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de loi de Hebb.

❖ 1943 : McCulloch et Pitts laissent leurs noms à une modélisation du neurone biologique (un neurone au comportement binaire). Ce sont les premiers à présenter le premier neurone formel et à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes (tout au moins au niveau théorique). Présentent et premier neurone formel.

❖ 1949 : D. Hebb, physiologiste américain propose un mécanisme d'apprentissage et il a expliqué le conditionnement chez l'animal par les propriétés du neurone eux-mêmes.

❖ 1958 : Rosenblatt présente les premiers réseaux de neurones artificiels : le perceptron. Il est inspiré du système visuel, et possède deux couches de neurone « perceptif et décisionnel ». Il a construit le premier neuro-ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance de formes. Notons qu'à cet époque les moyens à sa disposition sont limités et c'est une prouesse technologique que de réussir à faire fonctionner correctement cette machine plus de quelques minutes.

❖ 1960 : B. Widrow, un automaticien, développe le modèle Adaline (Adaptive Linear Element). Dans sa structure, le modèle ressemble au perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétro-propagation de gradient très utilisé aujourd'hui avec les perceptrons multicouches. Les réseaux de type Adaline restent utilisés de nos jours pour certaines applications particulières. B. Widrow a créé dès cette époque une des premières firmes proposant neuro-ordinateurs et neuro-composants, la « Memistor Corporation ».

❖ 1969 : M. Minsky et S. Papert publient un ouvrage qui met en exergue les limitations théoriques du perceptron. Limitations alors connues, notamment concernant l'impossibilité de traiter par ce modèle des problèmes non linéaires.

Limites du perceptron ; besoin d'architectures plus complexes, mais comment effectuer leur apprentissage.

❖ 1967-1982 : toutes les recherches ne sont, bien sûr, pas interrompues. Elles se poursuivent, mais déguisées, sous le couvert de divers domaines comme : le traitement adaptatif du signal, la reconnaissance de formes, la modélisation en neurobiologie, etc. Les grands noms travaillent durant cette période tels : S.Grossberg, T.Kohonen.

❖ 1982 : J.J.Hopfield est un physicien reconnu à qui l'on doit le renouveau d'intérêt pour les réseaux de neurones artificiels. A cela plusieurs raisons : Au travers d'un article court, clair et bien écrit, il présente une théorie du fonctionnement et des possibilités des réseaux de neurones. Il faut remarquer la présentation anticonformiste de son article. Alors que les auteurs s'acharnent jusqu'alors à proposer une structure et une loi d'apprentissage, puis à étudier les propriétés émergentes ; J.J.Hopfield fixe préalablement le comportement à atteindre pour son modèle et construit à partir de là, la structure et la loi d'apprentissage correspondant au résultat escompté. Ce modèle est aujourd'hui encore très utilisé pour des problèmes d'optimisation. D'autre part, entre les mains de ce physicien distingué, la théorie des réseaux de neurones devient respectable. Elle n'est plus l'apanage d'un certain nombre de psychologues et neurobiologistes hors du coup. Enfin, une petite phrase, placée en commentaire dans son article initial, met en avant l'isomorphisme de son modèle avec le modèle d'Ising (modèle des verres de spins). Cette idée va drainer un flot de physiciens vers les réseaux de neurones artificiels. Notons qu'à cette date, l'IA est l'objet d'une certaine désillusion, elle n'a pas répondu à toutes les attentes et s'est même heurtée à de sérieuses limitations. Aussi, bien que les limitations du perceptron mise en avant par M.Minsky ne soient pas levées par le modèle d'Hopfield, les recherches sont relancées.

❖ 1983 : la machine de Boltzmann est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables).

❖ 1986 : rétro propagation (Rumelhart et McClelland) Rumelhart popularise l'algorithme de rétro propagation du gradient, conçu par Werbos, Dès cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables, cet algorithme permet d'entraîner les couches cachées des réseaux multicouches. Nouvelles architectures de réseaux de neurones applications reconnaissance de l'écriture reconnaissance et synthèse de la parole vision (traitement d'images).

❖ 1990 : Société de L'information nouvelles application recherche et filtrage d'information sur le web extraction d'information, veille technologique multimédia (indexation, ...) datamining besoin de combiner plusieurs modèles [1.5].

Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques.

3.3 Du neurone biologique au neurone artificiel

3.3.1 Analogie avec le cerveau

Le fonctionnement exact du cerveau humain reste un mystère. Pourtant, certains aspects de ce processeur étonnant sont connus.

En particulier, l'élément le plus fondamental du cerveau humain est un type spécifique de cellule qui, contrairement au reste du corps, ne semble pas se régénérer. Ce type de cellule est la seule partie du corps qui n'est pas remplacé lentement ; on suppose que ces cellules sont celles qui nous donnent la capacité de nous souvenir, de réfléchir et d'appliquer les expériences à chacune de nos actions.

Ces cellules, toutes au nombre de 100 milliards, sont connus sous le nom de neurones. Chacun de ces neurones peut se connecter avec jusqu'à 200 000 autres neurones, bien que 1 000 à 10 000 soient typiques.

La puissance de l'esprit humain vient du nombre de ces composants de base et les multiples connexions entre eux. Il vient aussi de la programmation génétique et de l'apprentissage.

Les neurones individuels sont compliqués. Ils ont une myriade de pièces, sous-systèmes et mécanismes de contrôle. Ils véhiculent des informations via un hôte de voies électrochimiques. Il existe plus d'une centaine de classes différentes de neurones, selon la méthode de classification utilisée. Ensemble ces neurones et leurs connexions forment un processus qui n'est pas binaire, pas stable, et non synchrone.

Ces réseaux de neurones artificiels tentent de reproduire uniquement les éléments les plus fondamentaux de cet organisme complexe, polyvalent et puissant. Ils le font en une manière primitive. Mais pour l'ingénieur logiciel qui essaie de résoudre les problèmes, l'informatique neuronale n'a jamais consisté à reproduire des cerveaux humains. Il se concentre sur les machines et une nouvelle façon de résoudre les problèmes [1.6]

3.3.2 Aspects historiques

Dès 1943, Mac Culloch et Pitts ont proposé des neurones formels mimant les neurones biologiques et capables de mémoriser des fonctions booléennes simples. Les réseaux de neurones artificiels réalisés à partir de ce type de neurones sont ainsi inspirés du système nerveux.

Ils sont conçus pour reproduire certaines caractéristiques des mémoires biologiques par le fait qu'ils soient :

- ✓ Massivement parallèles ;
- ✓ Capables d'apprentissage ;
- ✓ Capable de mémoriser l'information dans les connexions inter-neurones ;
- ✓ Capables de traiter des informations incomplètes.

3.3.3 Définition

Un neurone, ou une cellule nerveuse, est une cellule excitable constituant l'unité fonctionnelle de la base du système nerveux.

Les neurones assurent la transmission d'un signal bioélectrique appelé influx nerveux. Ils ont deux propriétés physiologiques :

- ✓ l'excitabilité, c'est-à-dire la capacité de répondre aux stimulations et de convertir celles-ci en impulsions nerveuses ;
- ✓ et la conductivité, c'est-à-dire la capacité de transmettre les impulsions.

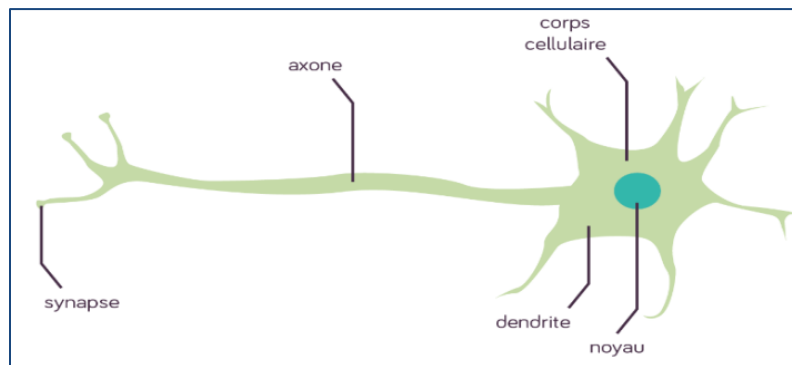


Figure 3-1 : Représentation d'un neurone biologique.

3.3.4 Structure des neurones

Le neurone est constitué de :

- Un corps de cellule, appelé le soma, c'est la partie centrale d'un neurone contenant le noyau cellulaire, qui fait la somme des influx qui lui parviennent. Si cette somme dépasse un certain seuil, il envoie lui-même un influx par l'intermédiaire de l'axone.
- Un nombre de fibres appelés dendrites qui sont les récepteurs principaux du neurone, captant les signaux qui lui parviennent.
- Une fibre longue appelé axone qui permet de transmettre les signaux émis par le corps cellulaire aux autres neurones.
- Les synapses qui permettent aux neurones de communiquer avec les autres via les axones et les dendrites.

Les méthodes connexionnistes ont été initialisées à l'ère de cybernétique. L'objectif des chercheurs était de construire une machine capable de reproduire le plus fidèlement possible certains aspects de l'intelligence humaine.

3.4 Neurones artificiels

3.4.1 Définition

C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés.

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires (neurone) fonctionnant en parallèle. Chaque neurone est doté d'un état interne, l'activation, par laquelle il influence les autres neurones du réseau, cette activation se propage dans le réseau le long des liens synaptiques. La règle qui détermine l'activation d'un neurone en fonction de l'influence de ces pairs est appelée fonction d'activation.

L'intérêt d'un réseau de neurones est le calcul d'une valeur de sortie unique par chaque neurone de sortie sur la base des informations qu'il reçoit.

Les réseaux de neurones artificiels sont des modèles qui peuvent être décrits par leurs comportements, leurs variables descriptives et les interactions des composants [2.13].

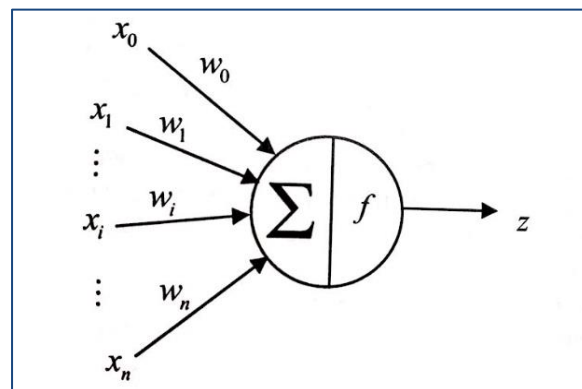


Figure 3-2 : Modèle de neurones artificiels de Mac Culloch et Pitts.

Chaque neurone artificiel transforme l'ensemble des signaux qu'il reçoit en un signal de sortie qui est communiqué à d'autres neurones. Cette transformation s'effectue en deux étapes :

- 1) le neurone effectue une sommation pondérée des potentiels (principe de superposition) ; la valeur numérique obtenue représente l'état du neurone qui l'a émis, afin d'obtenir une stimulation résultante globale :

$$Y = \sum w_i x_i \quad (3.1)$$

- 2) à l'aide d'une fonction de transfert, on teste le neurone. Si cette stimulation dépasse un certain seuil, le neurone est activé et transmet une réponse.

$$\text{Dans ce cas : } z = f(y) \quad (3.2)$$

Donc la formule mathématique sera de la forme :
$$\begin{cases} Y = \sum w_i x_i \\ z = f(y) \end{cases} \quad (3.3)$$

Telle que :

— Les coefficients de pondération w_i , $i = 0,1,2, \dots n$ s'appellent les poids synaptiques (w_i vecteur de poids synaptiques); Si w_i est positif, l'entrée x_i est excitatrice alors que si w_i est négatif, elle est inhibitrice [2.12].

— Et les x_i , $i = 0,1,2, \dots n$ s'appellent les entrées (x_i vecteur d'entrée).

3.4.2 Comportement

Le vecteur d'entrées x_i est multiplié par le vecteur des poids synaptiques, ce produit est sommé au niveau du neurone. Le neurone ne renvoi qu'une valeur.

A partir de cette valeur, une fonction de d'activation calcule la valeur de l'état du neurone, la sortie du neurone est donc $z = f(y) = f(\sum w_i x_i)$. C'est cette valeur qui sera transmise aux neurones aval.

Il existe de nombreuses formes possibles pour la fonction d'activation (Figure 3-3). On remarquera qu'à la différence des neurones biologiques dont l'état est binaire, la plus part des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle $[0, +1]$ ou $[-1, +1]$ selon la fonction utilisée [2.13].

3.4.3 Fonction d'activation

Dans le neurone de Mac Culloch et Pitts (Figure 3-2), la fonction d'activation f est du type tout ou rien à seuil prenant les valeurs 0 ou 1. Le seuil de déclenchement est en général provoqué par une entrée inhibitrice x_0 , parfois appelée biais.

Les fonctions d'activation les plus utilisées sont les suivantes :

- a) Tout ou rien ;
- b) Fonction signe ;
- c) Plus ou moins à seuil ;
- d) Fonction affine ;
- e) Saturation ;
- f) Sigmoides $F(x) = \frac{1}{1+e^{-x}}$;
- g) Fonction arc tangente ;
- h) Fonction radiale de base du type gaussien [1.7].

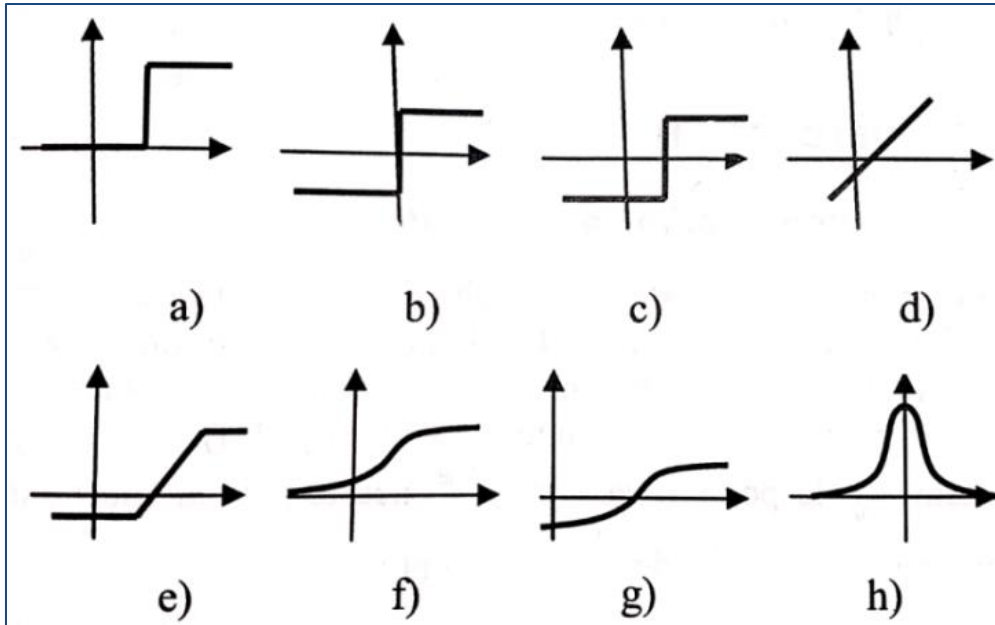


Figure 3-3 : Les fonctions d'Activation d'un réseau de neurone artificiel.

Tableau 3-1 : Equivalence entre le neurone biologique et le neurone artificiel.

Réseau de neurones biologiques	Réseau de neurones artificiels
Soma	Neurone
Dendrite	Entrée
Axone	Sortie
Synapse	Poids

3.4.4 L'architecture des réseaux de neurones

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau. [2.11] page30.

La couche la plus à gauche de ce réseau (Figure 3-4) est appelée couche d'entrée et les neurones de la couche sont appelés neurones d'entrée. La couche la plus à droite ou de sortie contient les neurones de sortie. La couche intermédiaire est appelée couche cachée, car les neurones de cette couche ne sont ni des entrées ni des sorties. Certains réseaux ont plusieurs couches cachées.

Selon le type de connexion, on distingue deux types de réseaux de neurones : les réseaux bouclés et non bouclés.

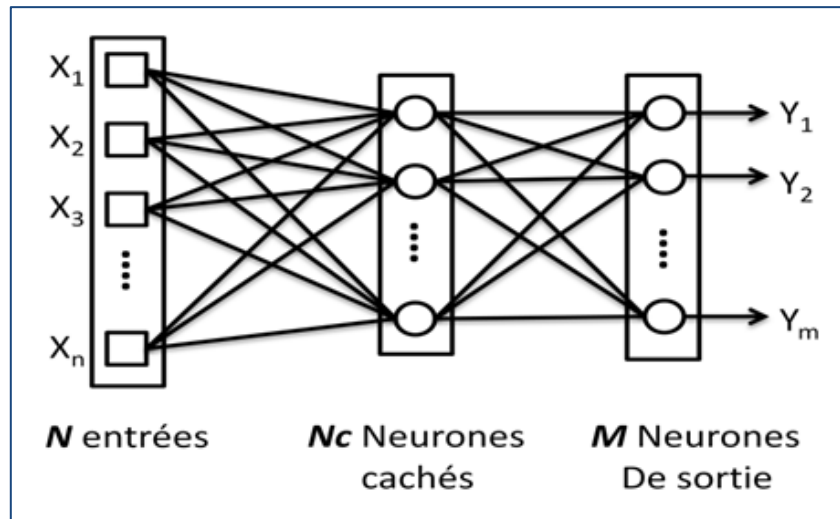


Figure 3-4 : Architecture d'un réseau de neurones à N entrées, une couche de N_c neurones cachés et M neurones de sortie.

3.4.4.1 Réseaux de neurones non bouclés

Dans un réseau de neurones non bouclés, le flux d'information circule des entrées vers les sorties sans « retour en arrière ». Les neurones sont regroupés par couches, et selon la tâche de chaque couche, on peut distinguer trois types de couches :

- Les neurones de la couche d'entrée : ce sont les neurones placés aux entrées du réseau, appelés aussi cellules perceptives du fait de leur propriété à acquérir des données en provenance de l'extérieur du système.
- Les neurones de la couche de sortie : ce sont les neurones qui effectuent les dernières opérations et diffusent les résultats du traitement effectués dans le système.
- Les neurones de la couche cachée : elles correspondent aux neurones, placés entre les neurones d'entrée et les neurones de sortie. Elles sont en nombre variable.

Ces différents neurones sont très souvent organisés par couches. Les neurones de sortie appartiennent à la couche de sortie, et les neurones cachés s'organisent en une ou plusieurs couches appelées couches cachées (Figure 3.4). Par ailleurs, les connexions entre deux couches, peuvent être partielles (locales) ou totales (Figure 3.5). L'utilisation de connexions partielles permet de relier un neurone à quelques neurones localisés dans son entourage pour effectuer une fonction spécifique [2.11].

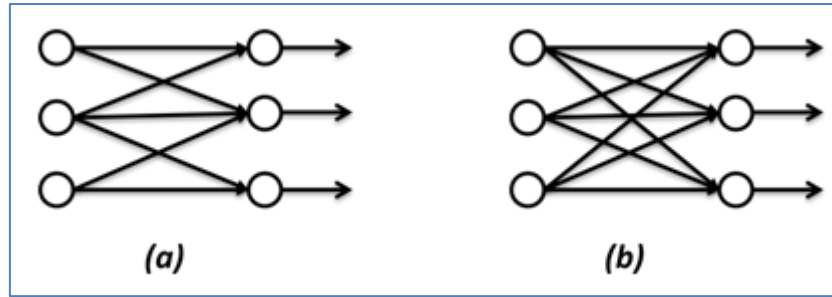


Figure 3-5 : Connexions : (a) partielles et (b) totales.

3.4.4.2 Réseaux de neurones bouclés

Il existe plusieurs types de réseaux de neurones bouclés. Le point commun entre tous ces réseaux est la présence d'au moins un chemin partant et revenant au même neurone (ce type de chemin est appelé « cycle »). Cela signifie que la sortie d'un neurone du réseau peut donc être fonction d'elle-même. Cela n'est certainement concevable que si la notion du temps est prise en considération.

Ainsi, chaque connexion d'un réseau de neurones bouclé est attachée avec un retard correspondant à un multiple entier de l'unité de temps choisie (éventuellement nul). À un instant donné, une grandeur ne pouvant pas être fonction de sa propre valeur au même instant, tout cycle du graphe du réseau doit avoir un retard non nul. Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locales.

Pour remédier au problème de la détermination de l'état du réseau par bouclage, on introduit sur chaque connexion « en retour » un retard qui permet de conserver le mode de fonctionnement séquentiel du réseau (Figure 3-6) [2.11].

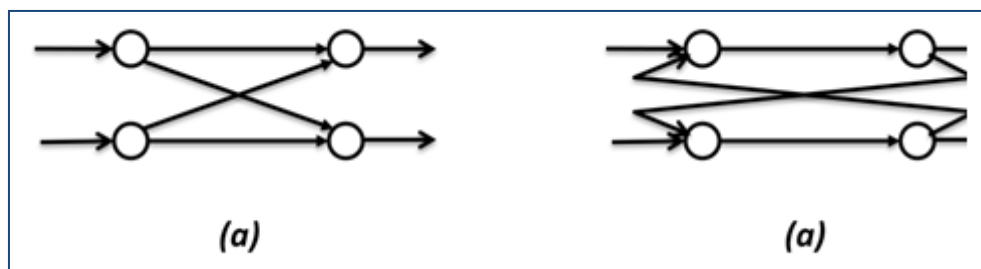


Figure 3-6 : Connexions (a) directes et (b) récurrentes.

3.5 Réseaux de neurones multicouche

« Le perceptron multicouche »

Il existe des types de réseaux de neurones à une seule couche comme le perceptron simple et l'Adaline².

Le premier réseau de neurones que nous allons étudier s'appelle le perceptron multicouche.

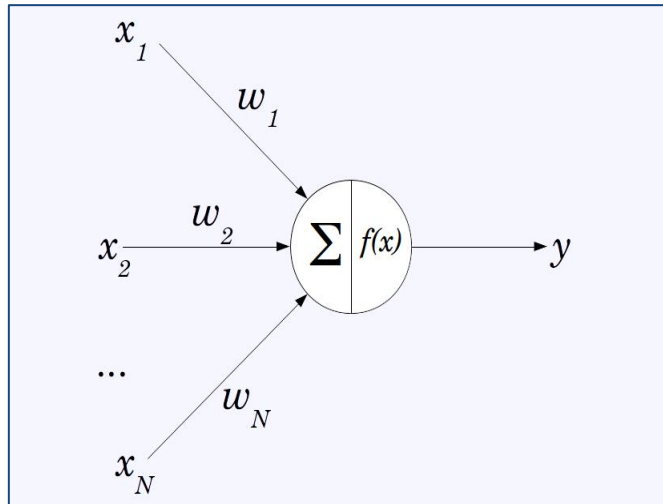


Figure 3-7 : Réseaux de neurones a une seule couche « le perceptron simple ».

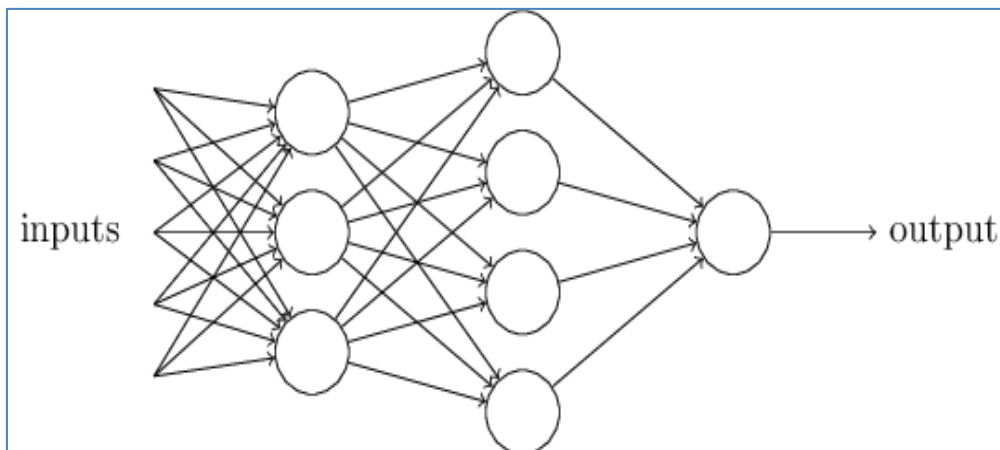


Figure 3-8 : Réseaux de neurones multicouche « le perceptron multicouche ».

² L'Adaline est un réseau de neurones avec une fonction d'activation sigmoïde.

3.5.1 Définition

Le perceptron multicouche est un des réseaux de neurones les plus utilisés pour des problèmes d'approximation, de classification et de prédiction. Il est habituellement constitué de deux ou trois couches de neurones totalement connectés.

Ce type de réseau est dans la famille générale des réseaux à propagation vers l'avant c'est-à-dire qu'en mode normal d'utilisation, l'information se propage dans un sens unique, des entrées vers les sorties sans aucune rétroaction. Son apprentissage est de type supervisé. Dans ce cas uniquement, le signal d'erreur est «rétro propagé» vers les entrées pour mettre à jour les poids des neurones.

3.5.2 Fonctionnement

Un perceptron prend plusieurs entrées binaires x_1, x_2, \dots , et produit une seule sortie binaire (Figure 3-8). Rosenblatt a proposé une règle simple pour calculer la sortie.

Il a introduit des poids w_1, w_2, \dots des nombres réels exprimant l'importance des entrées respectives par rapport à la sortie. La sortie du neurone, 0 ou 1, est déterminée par le fait que la somme pondérée $\sum_j w_j x_j$ est inférieure ou supérieure à une certaine valeur de seuil. Tout comme les poids, le seuil est un nombre réel qui est un paramètre du neurone.

Pour le dire en termes algébriques plus précis :

$$Output = \begin{cases} 0 & \text{si } \sum w_i x_i \leq \text{le seuil} \\ 1 & \text{si } \sum w_i x_i > \text{le seuil} \end{cases} \quad (3.4)$$

C'est le modèle mathématique de base. Une façon de penser au perceptron est qu'il s'agit d'un appareil qui prend des décisions en évaluant des preuves.

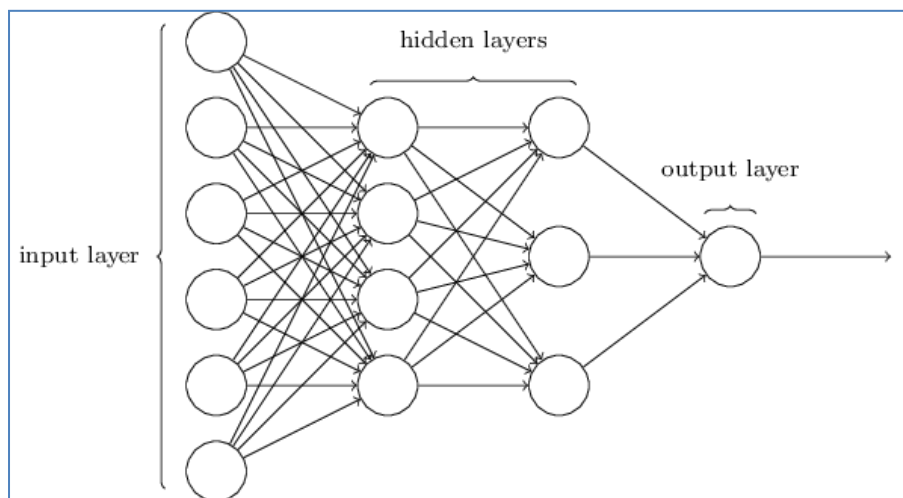


Figure 3-9 : Réseaux de neurones multicouche «le perceptron avec quatre couches».

Notion : _ La formule (3.4) peut s'écrire comme suivant : $\sum w_i x_i = w^t \cdot X$ (3.5)

_ Equivalente à l'écriture matricielle $W \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} \cdot X \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ (3.6)

_ Perceptron nulle : à partir de l'équation (3.4) on aura :

$$\text{Output} = \begin{cases} 0 & \text{si } w \cdot x + b \leq 0 \\ 1 & \text{si } w \cdot x + b > 0 \end{cases}, b = -\text{le seuil}.$$

3.6 Le neurone sigmoïde

3.6.1 Définition

Les neurones sigmoïdes sont similaires aux perceptrons, mais modifiés de sorte que de petits changements dans leur poids et leur biais ne provoquent qu'un petit changement dans leur sortie. C'est le fait crucial qui va permettre à un réseau de neurones sigmoïdes d'apprendre.

3.6.2 Fonctionnement

Nous allons représenter les neurones sigmoïdes de la même manière que nous avons représenté les perceptrons (Figure 3-7).

Tout comme un perceptron, le neurone sigmoïde a des entrées x_1, x_2, \dots . Mais au lieu d'être simplement 0 ou 1, ces entrées peuvent également prendre n'importe quelle valeur entre 0 et 1. Ainsi, par exemple, 0,638...0.887..., est une entrée valide pour un neurone sigmoïde. Aussi, tout comme un perceptron, le neurone sigmoïde a des poids pour chaque entrée w_1, w_2, \dots et un biais global b . Mais la sortie n'est pas 0 ou 1.

Au lieu de cela, c'est $\sigma(w \cdot x + b)$, où σ est appelée la fonction sigmoïde et est définie par :

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (3.7)$$

Pour le dire un peu plus explicitement, la sortie d'un neurone sigmoïde avec des entrées x_1, x_2, \dots des poids w_1, w_2, \dots et un biais b est :

$$\frac{1}{1+\exp(-\sum_j w_j x_j - b)} \quad (3.8)$$

Il existe de nombreuses similitudes entre les perceptrons et les neurones sigmoïdes, Pour comprendre la similitude avec le modèle perceptron, supposons que $z \equiv w \cdot x + b$ soit un grand nombre positif. Alors $e^{-z} \approx 0$ et donc $\sigma(z) \approx 1$;

En d'autres termes, lorsque, $z = w \cdot x + b$ est grand et positif, la sortie du neurone sigmoïde est d'environ 1, comme cela aurait été le cas pour un perceptron. Supposons par contre que $z = w \cdot x + b$ soit très négatif. Alors $e^{-z} \rightarrow \infty$, et $\sigma(z) \approx 0$. Ainsi, lorsque, $z = w \cdot x + b$ est très négatif, le comportement d'un neurone sigmoïde se rapproche également étroitement d'un perceptron. Ce n'est que lorsque $w \cdot x + b$ est de taille modeste qu'il y a beaucoup d'écart par rapport au modèle perceptron.

En fait, la forme exacte de σ n'est pas si importante, ce qui compte vraiment, c'est la forme de la fonction lorsqu'elle est tracée (représentation du graphe [Figure 3-10](#)) :

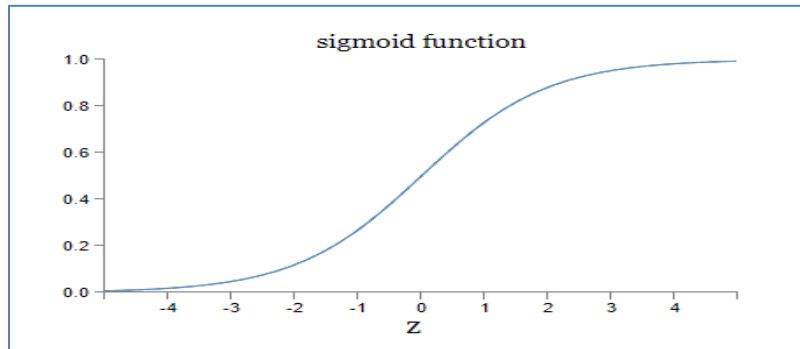


Figure 3-10 : Représentation de la fonction sigmoïde.

Cette forme est une version lissée d'une fonction en escalier :

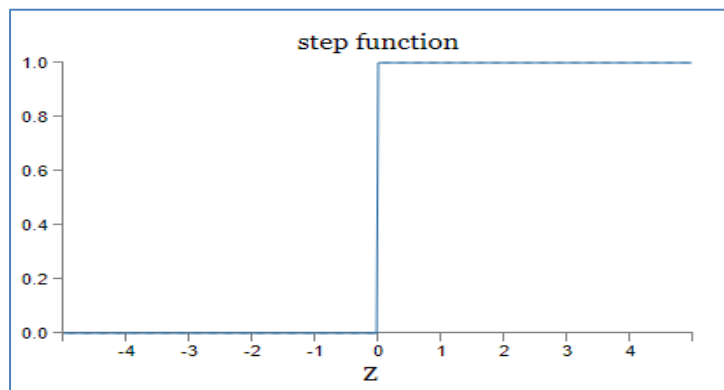


Figure 3-11 : Représentation de la fonction en escalier.

Si σ avait en fait été une fonction en escalier, alors le neurone sigmoïde serait un perceptron, puisque la sortie serait 1 ou 0 d'après $w \cdot x + b$ s'il était positif ou négatif.

En utilisant la fonction σ réelle, nous obtenons, comme déjà sous-entendu ci-dessus, un perceptron lissé. En effet, c'est la régularité de la fonction σ qui est le fait crucial, pas sa forme détaillée. La régularité de σ signifie que de petits changements Δw_j dans les poids et Δb dans le biais produiront un petit changement $\Delta output$ dans la sortie du neurone.

En fait, le calcul nous dit que $\Delta output$ est bien approché par :

$$\Delta output \approx \sum_j \frac{\partial output}{\partial w_j} \Delta w_j + \frac{\partial output}{\partial b} \Delta b \quad (3.9)$$

Où la somme est sur tous les poids w_j , $\frac{\partial output}{\partial w_j}$ et $\frac{\partial output}{\partial b}$ dénotent les dérivées partielles de la sortie par rapport à w_j et b respectivement. $\Delta output$ est une fonction linéaire des changements Δw_j et Δb dans les poids et les biais.

Cette linéarité facilite le choix de petits changements dans les poids et les biais pour obtenir tout petit changement souhaité dans la sortie. Ainsi, bien que les neurones sigmoïdes aient à peu près le même comportement qualitatif que les perceptrons, ils permettent de comprendre beaucoup plus facilement comment la modification des poids et des biais modifiera la sortie.

Si c'est la forme de σ qui compte vraiment, et non sa forme exacte, alors pourquoi utiliser la forme particulière utilisée pour σ dans l'équation :

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (3.10)$$

Une grande différence entre les perceptrons et les neurones sigmoïdes est que les neurones sigmoïdes ne se contentent pas de sortir 0 ou 1. Ils peuvent avoir comme sortie n'importe quel nombre réel entre 0 et 1, donc des valeurs telles que 0,173... et 0,689... sont des sorties légitimes.

3.6.3 Comme fonction d'activation dans les réseaux de neurones

La fonction sigmoïde est utilisée comme fonction d'activation dans les réseaux de neurones.

Juste pour revoir ce qu'est une fonction d'activation, la (Figure 3.12) ci-dessous montre le rôle d'une fonction d'activation dans une couche d'un réseau de neurones.

Une somme pondérée d'entrées passe par une fonction d'activation et cette sortie sert d'entrée à la couche suivante.

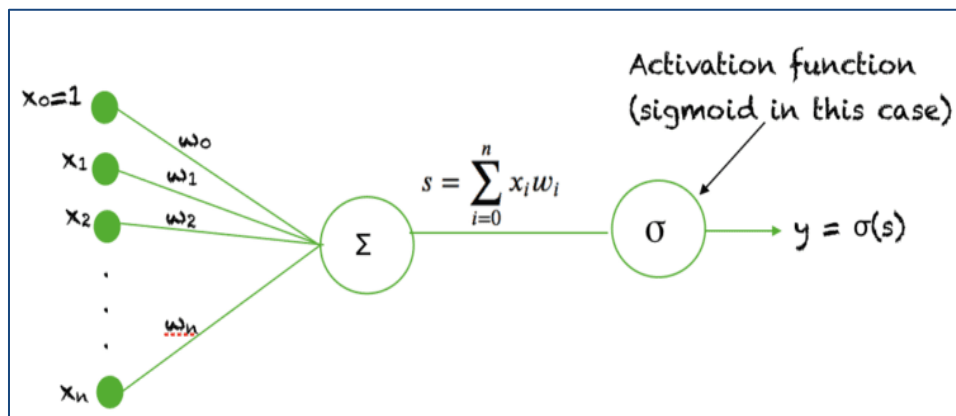


Figure 3-12 : Fonction d'activation dans une couche d'un réseau de neurones.

Lorsque la fonction d'activation d'un neurone est une fonction sigmoïde, c'est une garantie que la sortie de cette unité sera toujours comprise entre 0 et 1. De plus, comme le sigmoïde est une fonction non linéaire, la sortie de cette unité serait une non-fonction linéaire de la somme pondérée des entrées. Un tel neurone qui utilise une fonction sigmoïde comme fonction d'activation est appelé unité sigmoïde.

3.6.4 L'importance de la fonction sigmoïde dans les réseaux de neurones

Si nous utilisons une fonction d'activation linéaire dans un réseau de neurones, alors ce modèle ne peut apprendre que des problèmes linéairement séparables. Cependant, avec l'ajout d'une seule couche cachée et d'une fonction d'activation sigmoïde dans la couche cachée, le réseau de neurones peut facilement apprendre un problème non linéairement séparable. L'utilisation d'une fonction non linéaire produit des limites non linéaires et, par conséquent, la fonction sigmoïde peut être utilisée dans les réseaux de neurones pour l'apprentissage de fonctions de décision complexes.

La seule fonction non linéaire qui peut être utilisée comme fonction d'activation dans un réseau de neurones est celle qui augmente de façon monotone. Ainsi, par exemple, $\sin(x)$ ou $\cos(x)$ ne peuvent pas être utilisés comme fonctions d'activation. Aussi, la fonction d'activation doit être définie partout et doit être continue partout dans l'espace des nombres réels. La fonction doit également être dérivable sur tout l'espace des nombres réels.

Typiquement, un algorithme de rétro-propagation utilise la descente de gradient pour apprendre les poids d'un réseau de neurones. Pour dériver cet algorithme, la dérivée de la fonction d'activation est requise [2.14].

Le fait que la fonction sigmoïde soit monotone, continue et dérivable partout, couplé à la propriété que sa dérivée peut être exprimée en termes d'elle-même, permet de dériver facilement les équations de mise à jour pour apprendre les poids dans un réseau de neurones lors de l'utilisation de l'algorithme de rétro-propagation.

3.7 Les règles d'apprentissage des réseaux de neurones

Un réseau de neurones artificiels de la règle d'apprentissage ou le processus d'apprentissage est une méthode, la logique mathématique ou algorithme qui améliore les performances et/ou le temps de formation du réseau. Habituellement, cette règle est appliquée à plusieurs reprises sur le réseau.

Cela se fait en mettant à jour les poids et les niveaux de biais d'un réseau lorsqu'un réseau est simulé dans un environnement de données spécifique [1.8].

Une règle d'apprentissage peut accepter les conditions existantes (pondérations et biais) du réseau et comparera le résultat attendu et le résultat réel du réseau pour donner des valeurs nouvelles et améliorées pour les pondérations et les biais. Selon la complexité du modèle réel simulé [1.9].

La règle d'apprentissage du réseau peut être aussi simple qu'une porte XOR ou une erreur quadratique moyenne, ou aussi complexe que le résultat d'un système d'équations différentielles.

La règle d'apprentissage est l'un des facteurs qui décide de la vitesse ou de la précision du développement du réseau artificiel. Selon le processus de développement du réseau, on a déjà parlé des modèles d'apprentissage dans notre deuxième chapitre.

De nombreuses méthodes d'apprentissage de l'apprentissage automatique fonctionnent de la même manière et sont basées les unes sur les autres, ce qui rend difficile leur classification dans des catégories claires.

Pour nous les méthodes d'apprentissage qui nous intéressent sont les suivantes :

- ✓ Règle de Hebb ;
- ✓ Règle de WIDROW HOFF.

3.7.1 Règle de Hebb

La règle de Hebb, a été proposée par Donald O Hebb, en 1949.

Elle indique que lorsque deux neurones sont excités en même temps, il faut modifier les coefficients synaptiques pour renforcer cette excitation simultanée.

C'est l'une des premières règles d'apprentissage et aussi les plus faciles dans le réseau de neurones. Elle est utilisée pour la classification des modèles. C'est un réseau de neurones à couche unique, c'est-à-dire qu'il a une couche d'entrée et une couche de sortie. La couche d'entrée peut avoir plusieurs unités. La couche de sortie n'a qu'une seule unité.

La règle de Hebb fonctionne en mettant à jour les poids entre les neurones du réseau neuronal pour chaque échantillon d'apprentissage.

Elle est à la fois utilisée comme hypothèse en neurosciences et comme concept dans les réseaux neuronaux en mathématiques. C'est une règle d'apprentissage des réseaux de neurones artificiels dans le contexte de l'étude d'assemblées de neurones. Elle décrit les changements d'adaptation neuronale dans le cerveau ou dans un réseau de neurones pendant un processus d'apprentissage.

Mise à jour des points (biais et poids) on appliquant la règle de Hebb :

$$\boxed{w'_i = w_i + n(y \cdot x_i)} \quad \text{Et} \quad \boxed{b'_i = b_i + n(y \cdot x_i)} \quad (3.11)$$

Avec :

- w'_i : les poids i corrigé ;
- w_i : les poids actuel ;
- b'_i : les biais i corrigé ;
- b_i : les biais actuel ;
- n : le taux d'apprentissage ;
- y : la sortie observée ;
- x_i : l'entrée du poids i pour la sortie attendue.

3.7.2 Règle de Widrow Hoff

Établie en 1960, il indique que la modification de ses coefficients est proportionnelle à l'erreur entre le résultat souhaité et le résultat réel et aux valeurs d'entrée. La phase d'apprentissage consiste à modifier les poids jusqu'à obtention d'une stabilisation du réseau. C'est-à-dire, jusqu'à ce que le poids ne se modifie plus que d'une façon minimale. L'apprentissage mathématique, basé sur ce concept, sert à minimiser une fonction de coûts formulée autour de l'erreur de sortie. Alors l'adaptation commence, par les neurones de la couche de sortie, forcés de la bonne valeur, puis on fait varier légèrement les poids des neurones des couches précédentes.

3.8 L'algorithme d'apprentissage standard pour les réseaux de neurones

La rétro-propagation du gradient de l'erreur ou «back propagation» est un algorithme d'optimisation, permettant d'ajuster les paramètres d'un réseau de neurones multicouches pour mettre en correspondance des entrées et des sorties référencées dans une base d'apprentissage.

Pour pouvoir entraîner ces systèmes, il faut savoir comment ajuster les paramètres de chaque couche de neurones. La rétro propagation permet de calculer le gradient de l'erreur pour chaque neurone, de la dernière couche vers la première. Le calcul de ce gradient se fait par la méthode de rétro propagation, pratiquée depuis le milieu des années 80.

Cela permet de corriger les erreurs selon l'importance des éléments qui ont justement participé à la réalisation de ces erreurs. Ainsi, les poids synaptiques qui contribuent à engendrer une erreur importante se verront modifiés de manière plus significative que les poids qui ont engendré une erreur marginale. Moyennant quelques précautions lors de l'apprentissage, les procédures d'optimisation finissent par aboutir à une configuration stable, généralement un extremum local, au sein du réseau de neurones.

Tableau 3-2 : Résumé de la rétro-propagation du gradient.

<p><i>La structure de l'algorithme de rétro-propagation</i></p>	
<i>Type du réseau</i>	<i>Multicouches</i>
<i>Les couches</i>	<p><i>1 Couche d'entrée</i> <i>1 ou plusieurs couches cachées</i> <i>1 couche de sortie</i></p>
<i>Types d'entrées</i>	<p><i>Binaires</i> <i>Entières</i> <i>Réelles</i></p>
<i>Fonction d'activation</i>	<i>Sigmoïde</i>
<i>Type d'apprentissage</i>	<i>supervisé</i>
<i>Algorithme d'apprentissage</i>	<i>La rétro propagation du gradient</i>
<i>Utilisé principalement dans les domaines</i>	<p><i>Opérations logiques complexes</i> <i>Classification des modèles</i> <i>Analyse de la parole</i></p>

La fonction de coût en particulier "l'erreur quadratique" :

$$c(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2 \quad (3.12)$$

Telle que :

- ✓ w : désigne la collection de tous les poids du réseau ;
- ✓ b : les biais du réseau ;
- ✓ n : le nombre total d'entrées d'apprentissage ;
- ✓ a : la valeur sortie du réseau lorsque x est entré ;
- ✓ y : la valeur réelle.

Le but de notre algorithme d'apprentissage sera donc de minimiser le coût $c(w, b)$ en fonction des poids et des biais.

En d'autres termes, nous voulons trouver un ensemble de poids et de biais qui rendent le coût aussi faible que possible. Nous le ferons en utilisant un algorithme connu sous le nom de descente de gradient.

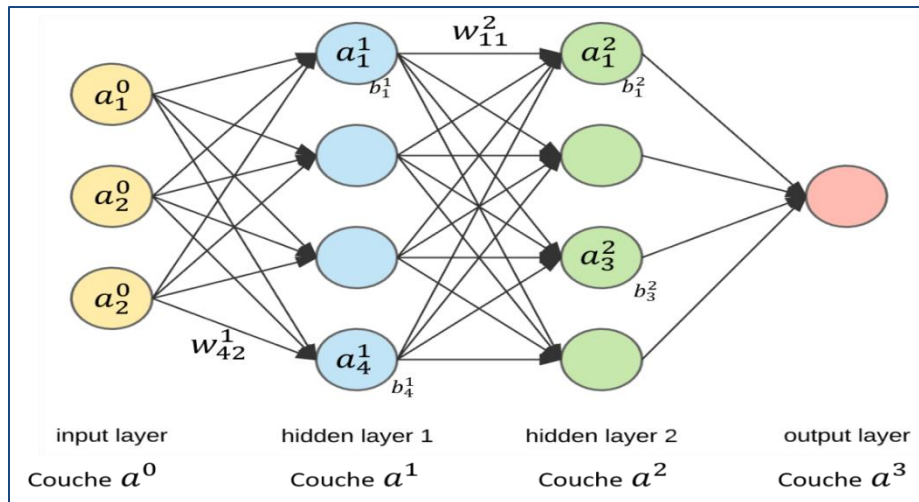


Figure 3-13 : Exemple d'une représentation de réseau de neurone.

La valeur d'activation du neurone j de la couche l :

On note $z_j^l = w_{jk}^l a_k^{l-1} + b_j^l$ la valeur d'agrégation du $j^{i\grave{e}me}$ neurone de la $l^{i\grave{e}me}$ couche, c'est-à-dire la valeur qu'un neurone calcule avant de la passer de la fonction d'activation.

a_j^l Est la valeur d'activation du $j^{i\grave{e}me}$ neurone de la $l^{i\grave{e}me}$ couche, c'est-à-dire la valeur définitive crachée par le neurone :

$$a_j^l = \sigma(\sum_k w_{jk}^l a_k^{l-1} + b_j^l) \tag{3.13}$$

Telle que :

- ✓ σ : la fonction d'activation sigmoïde.
- ✓ a_j^l : la valeur d'activation du neurone j de la couche l .
- ✓ b_j^l : La fonction de biais du neurone j de la couche l .
- ✓ w_{jk}^l : sont les poids de la connexion du neurone k de la couche $(l - 1)$ au neurone l de la couche l .
- ✓ k : Parcourt tous les neurones de la couche $l - 1$.

On remplace les notations matricielles $w^l = (w_{jk}^l)$; $b^l = (b_j^l)$; $a^l = (a_j^l)$ dans l'équation (3.13), on aura :

$$a^l = \sigma(w^l a^{l-1} + b^l) \tag{3.14}$$

D'où

$$\boxed{a^l = \sigma(z^l)} \tag{3.15}$$

A ce stade, nous n'avons fait que rappeler des concepts déjà exposés et des éléments de notation, maintenant on va aborder le fonctionnement de notre algorithme d'apprentissage d'un réseau de neurone.

Soit la répétition jusqu'à ce que $c(w, b)$ converge pour tout les poids et le biais :

$$\begin{cases} w_k \rightarrow w'_k = w_k - \eta \frac{\partial c}{\partial w_k} \\ b_l \rightarrow b'_l = b_l - \eta \frac{\partial c}{\partial b_l} \end{cases}$$

La seule difficulté est de calculer $\frac{\partial c}{\partial w}$, $\frac{\partial c}{\partial b}$, d'où notre but est de calculer $\frac{\partial c}{\partial w}$, $\frac{\partial c}{\partial b}$ pour tout $w \in W$ et $b \in B$.

On définit l'erreur δ_j^l comme suivant :

$$\delta_j^l = \frac{\partial c}{\partial z_j^l} \quad (3.16)$$

Pour la couche de sortie :

$$\delta_j^L = \frac{\partial c}{\partial a_j^L} \sigma'(z_j^L) \quad (3.17)$$

On écrit (3.17) sous forme matricielle ça va nous donner l'équation (3.18) :

$$\delta^L = \nabla_a c \odot \sigma'(z^L) \quad (3.18)$$

Si $c(w, b)$ est sous forme quadratique, alors (3.18) sera de la forme suivant :

$$\delta^L = (a^L - y) \odot \sigma'(z^L) \quad (3.19)$$

L'équation (3.16) de l'erreur δ^l en termes de l'erreur dans le niveau suivant δ^{l+1} :

$$\delta^l = [(w^{l+1})^t \cdot \delta^{l+1}] \odot \sigma'(z^l) \quad (3.20)$$

On donne l'équation de la dérivée du coût $c(w, b)$ par rapport au biais :

$$\frac{\partial c}{\partial b_j^l} = \delta_j^l \quad (3.21)$$

Aussi l'équation de la dérivée de $c(w, b)$ par rapport au poids w_{jk}^l :

$$\frac{\partial c}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \quad (3.22)$$

L'équation (3.21) sous forme matricielle :

$$\frac{\partial c}{\partial b} = \delta_{out}$$

L'équation (3.22) Sous forme matricielle :

$$\frac{\partial c}{\partial w} = a_{in} \delta_{out}$$

3.8.1 L'algorithme back propagation du gradient

La rétro propagation «back propagation » du gradient est une méthode pour entraîner un réseau de neurones, consistant à mettre à jour les poids de chaque neurone de la dernière couche vers la première, les poids synaptiques qui contribuent plus à une erreur seront modifiés de manière plus importante que les poids qui provoquent une erreur marginale.

Pour la mise en œuvre de l'apprentissage des réseaux par l'algorithme de back propagation « retro-propagation » du gradient, nous avons appliqué l'algorithme suivant les étapes :

Etape 1 : lecture de la valeur $x = a^1$ (activation de la couche d'entre).

Etape 2 : donner le nombre des cellules L .

Etape 3 : initialisation aléatoire de b^1 et w^1 .

Etape 4 : Pour $l=2, \dots, L$ calculer :

$$z^l = w^l a^{l-1} + b^l \text{ et } a^l = \sigma(z^l)$$

Etape 5 : Calcule de l'erreur de la sortie du réseau:

$$\delta^L = \nabla_a c \odot \sigma'(z^L)$$

Etape 6 : Calculer l'erreur de Rétro propagation: pour $l = L - 1, L - 2, \dots, 1$

$$\delta^l = [(w^{l+1})^t \delta^{l+1}] \odot \sigma'(z^l)$$

Etape 7 : résultat d'apprentissage (le gradient de la fonction):

$$\frac{\partial c}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \text{ et } \frac{\partial c}{\partial b_j^l} = \delta_j^l$$

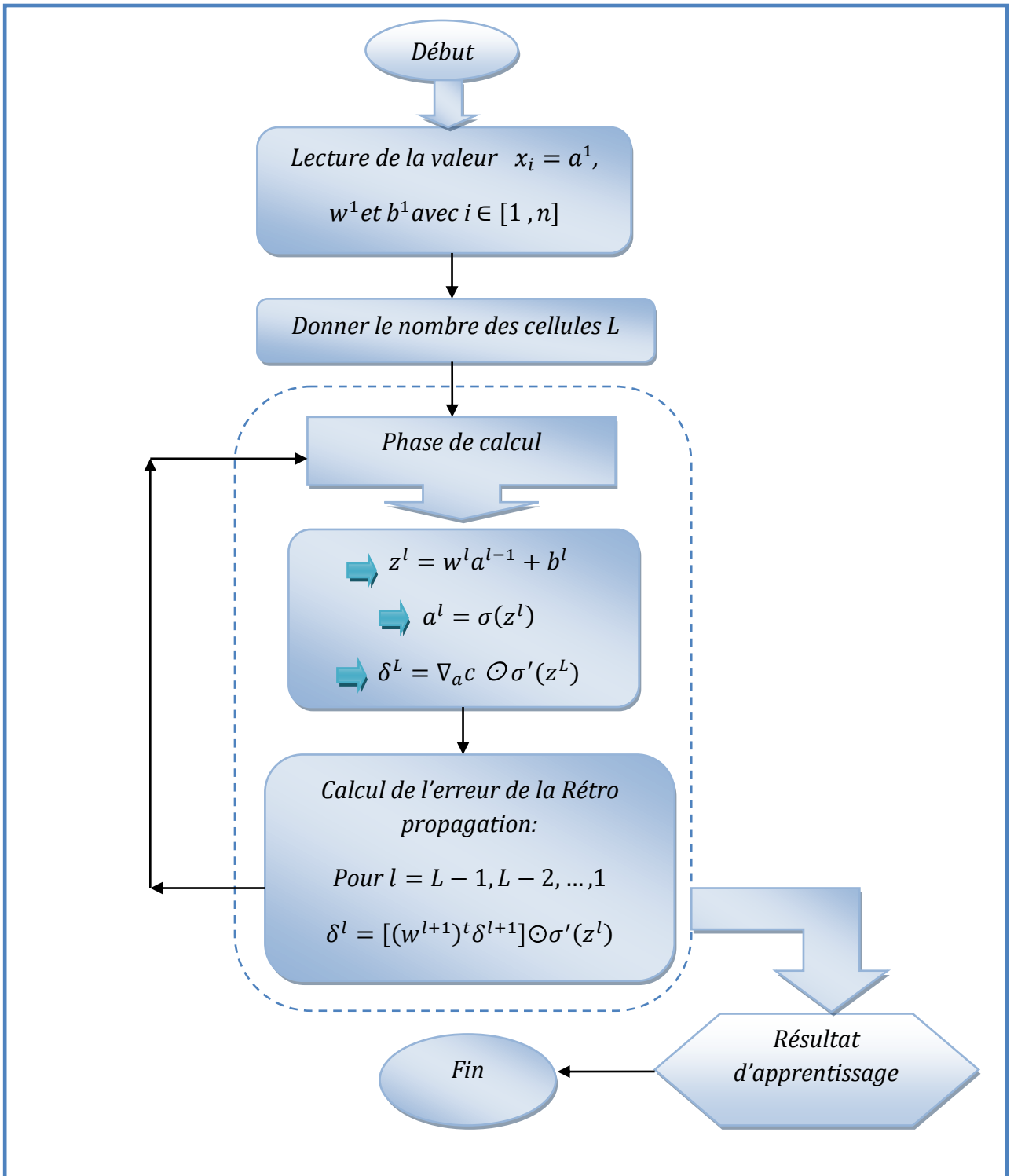


Figure 3-14 : Organisme de l'algorithme de rétro-propagation du gradient.

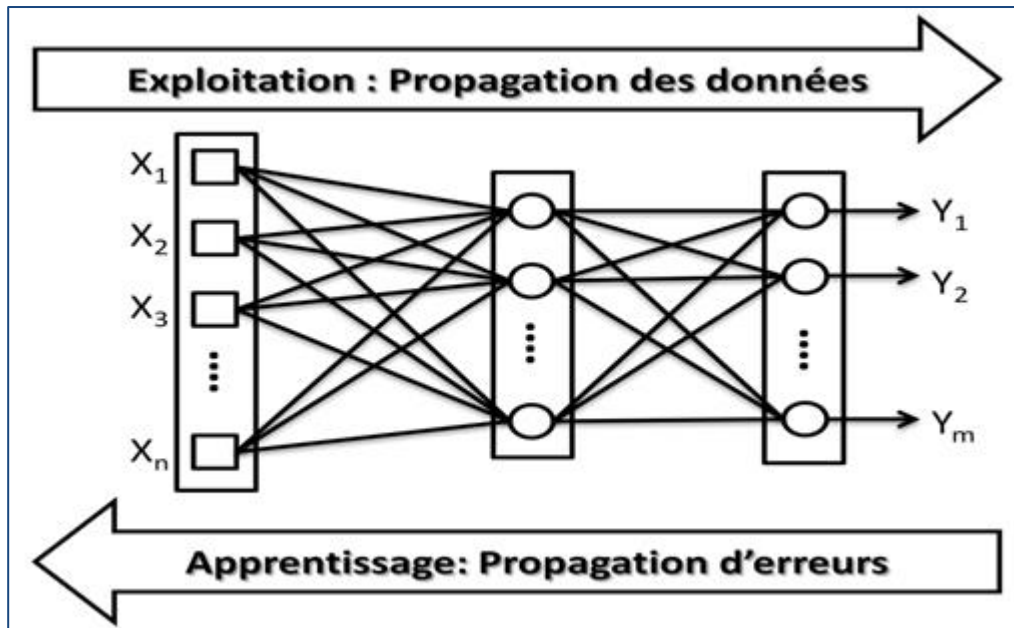


Figure 3-15 : Schéma de fonctionnement de l'algorithme de rétro propagation.

3.9 Conclusion

Un réseau de neurones artificiels, est un système informatique matériel et logiciel dont le fonctionnement est calqué sur celui des neurones du cerveau humain ; Il s'agit là d'une variété de technologie Deep Learning.

Pour entraîner un réseau de neurones, on utilise la rétro propagation du gradient consistant à mettre à jour les poids de chaque neurone de la dernière couche vers la première. Elle vise à corriger les erreurs selon l'importance de la contribution de chaque élément à celles-ci.

Chapitre 4 : La mémoire longue à court terme « LSTM »

« L'intelligence artificielle ne fait pas le poids

Face à la stupidité naturelle ».

ALBERT EINSTEIN

La plus grands partis de ce chapitre a adopté la référence [3.8].

4.1 Introduction

La mémoire longue à court terme (LSTM) est un réseau de neurones artificiels utilisé dans les domaines de l'intelligence artificielle et de l'apprentissage en profondeur. Contrairement aux réseaux de neurones à anticipation standard, LSTM a des connexions de rétroaction. Ils sont aussi bien adaptés à la classification, au traitement et à la réalisation de prédictions basées sur des données de séries chronologiques, car il peut y avoir des décalages de durée inconnue entre des événements importants dans une série chronologique.

Les LSTMs ont été développés pour traiter le problème du gradient de fuite qui peut être rencontré lors de la formation des RNN traditionnels.

Dans ce chapitre nous allons d'abord citer quelques notions des réseaux de neurones récurrents comme les LSTMs qui représente une cellule de se réseaux; nous parlerons également de sa structure et de son intérêt dans un réseau de neurone récurrent.

4.2 Réseaux de neurones récurrents

Les humains ne commencent pas leurs réflexions à partir de zéro à chaque seconde. En lisant cette phrase, vous comprenez chaque mot en fonction de votre compréhension des mots précédents. Vous ne jetez pas tout et recommencez à penser à partir de zéro. Vos pensées ont de la persistance.

Les réseaux de neurones traditionnels ne peuvent pas faire cela, et cela semble être une lacune majeure. Par exemple, imaginez que vous souhaitiez classer le type d'événement qui se produit à chaque instant d'un film. On ne sait pas comment un réseau neuronal traditionnel pourrait utiliser son raisonnement sur les événements précédents du film pour informer les événements ultérieurs. Les réseaux de neurones récurrents résolvent ce problème. Ce sont des réseaux avec des boucles qui permettent aux informations de persister.

4.2.1 Définition

Un réseau de neurones récurrents (RNN) est un réseau de neurones artificiel présentant des connexions récurrentes. Un réseau de neurones récurrent est constitué d'unités (neurones) interconnectées interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ses entrées.

Les réseaux de neurones récurrents sont adaptés pour des données d'entrée de taille variable. Ils conviennent en particulier pour l'analyse de séries temporelles. Ils sont utilisés en reconnaissance automatique de la parole ou de l'écriture manuscrite - plus en général en reconnaissance de formes - ou encore en traduction automatique. « Dépliés », ils sont comparables à des réseaux de neurones classiques avec des

contraintes d'égalité entre les poids du réseau. Les techniques d'entraînement du réseau sont les mêmes que pour les réseaux classiques (rétro propagation du gradient), néanmoins les réseaux de neurones récurrents se heurtent au problème de disparition du gradient pour apprendre à mémoriser des évènements passés. Des architectures particulières répondent à ce dernier problème, on peut citer en particulier les réseaux Long short term memory.

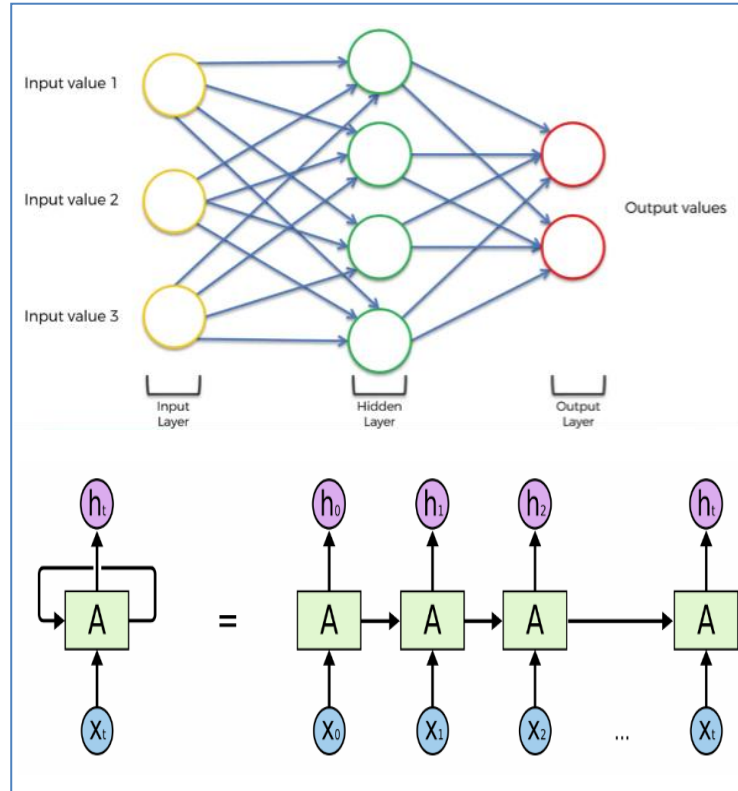


Figure 4-1 : Réseau de neurones récurrents.

On considère $A(0)$ comme étant la séquence d'entrée, puis on obtient en sortie un état h_0 du neurone. Puis $A(1)$ correspondra à l'entrée de l'étape suivante. De même, la sortie h_1 correspondra à l'entrée de $A(2)$ pour l'étape suivante et ainsi de suite. De cette façon, il se souvient du contexte pendant la formation. En termes simples, il y a une couche d'entrée, qui correspond probablement à une couche cachée avec certaines activations cela renvoie en sortie l'état obtenu.

La formule de l'état actuelle est donnée par :

$$h_t = f(h_{t-1}, x_t) \quad (4.1)$$

Avec :

- ✓ h_t : État actuel ;
- ✓ h_{t-1} : État précédent ;
- ✓ x_t : État d'entrée.

Puis, la formule généralement utilisée afin d'appliquer une fonction d'activation est donnée par :

$$h_t = \tanh(w_{hh}h_{t-1} + w_{xh}x_t) \quad (4.2)$$

Avec :

- ✓ w_{hh} : Poids du neurone récurrent ;
- ✓ w_{xh} : Poids du neurone d'entrée.

La formule pour calculer la sortie est donnée par :

$$y_t = w_{hy}h_t \quad (4.3)$$

Avec :

- ✓ y_t : Sortie du neurone ;
- ✓ w_{hy} : Poids de la couche de sortie.

4.2.2 Représentation

Les figures ci-dessous représentent les différentes représentations possibles d'un réseau de neurone récurrent

1. Écraser le réseau. Les couches sont toujours là, mais pensez-y comme si nous regardions sous ce réseau de neurones.

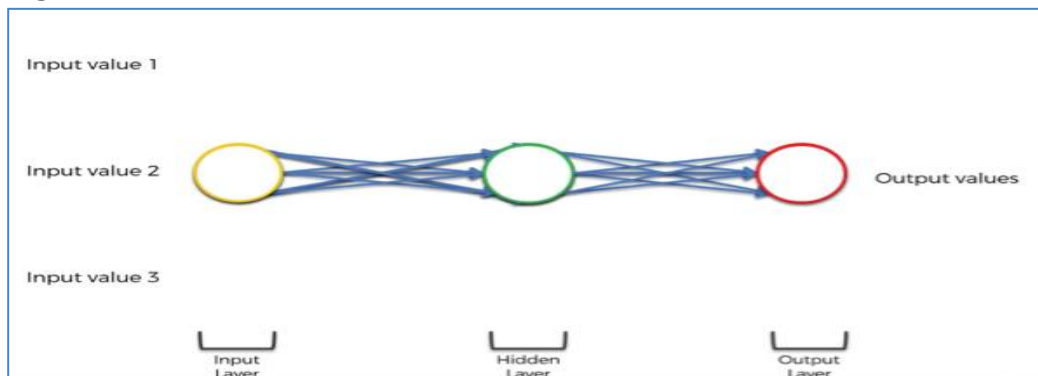


Figure 4-2 : Représentation N°1.

2. Changer les flèches multiples en deux.

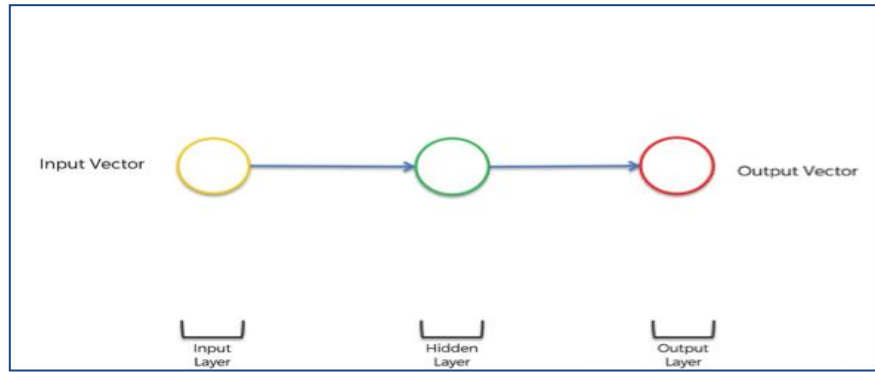


Figure 4-3 : Représentation N°2.

3. Le tordre pour le rendre vertical parce que c'est la représentation standard.

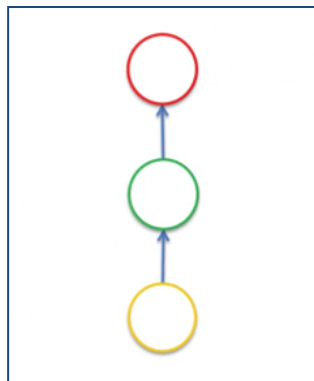


Figure 4-4 : Représentation N°3.

4. Ajout d'une ligne, qui représente une boucle temporelle. Il s'agit d'une représentation à l'ancienne des RNN et signifie essentiellement que cette couche cachée donne non seulement une sortie, mais se réinjecte également en elle-même.

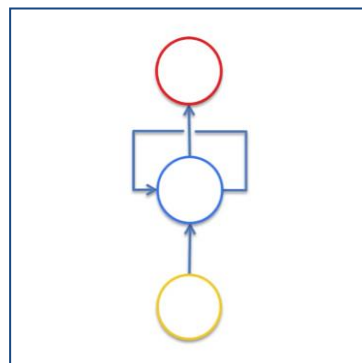


Figure 4-5 : Représentation N°4.

5. Dérouler la boucle temporelle et représenter les RNN d'une nouvelle manière. Maintenant, chaque cercle représente non seulement un neurone, mais toute une couche de neurones.

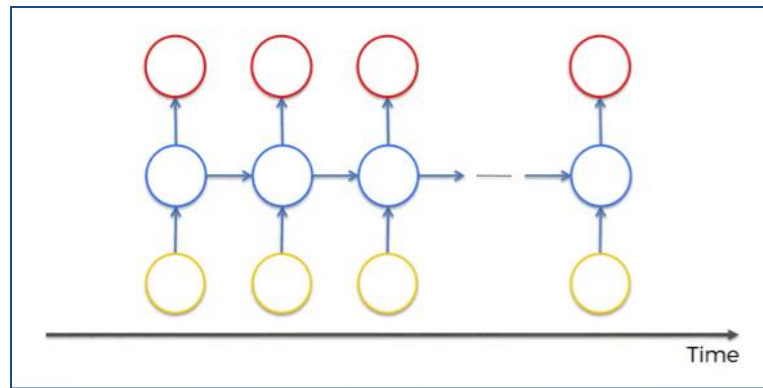


Figure 4-6 : Représentation N°5.

Donc, nous avons des entrées, une couche cachée et des sorties comme d'habitude, mais maintenant les neurones sont également connectés à eux-mêmes à travers le temps. L'idée derrière les RNN est que les neurones ont une sorte de mémoire à court terme leur offrant la possibilité de se souvenir de ce qui était dans ce neurone juste auparavant. Ainsi, les neurones peuvent se transmettre des informations dans le futur et analyser les choses.

4.2.3 Exemples d'application RNN

Voici cinq exemples d'application RNN :

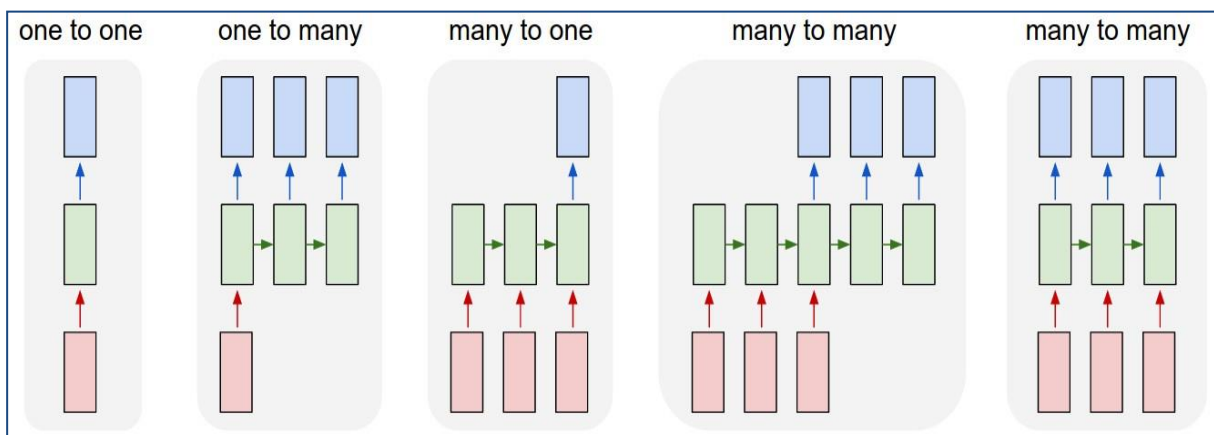


Figure 4-7 : Représentation de cinq exemples d'application RNN.

Prenons un exemple de plusieurs à un "many to one": Un exemple de cette relation serait l'analyse des sentiments, lorsque vous avez beaucoup de texte, comme le commentaire d'un client, par exemple, et que vous devez évaluer quelle est la probabilité que ce commentaire soit positif, ou à quel point ce commentaire est réellement positif, ou négatif.

Aussi pour plusieurs à plusieurs "many to many": Les traductions peuvent être un bon exemple de plusieurs types de réseau.

4.3 Le problème du gradient de fuite

4.3.1 Bref historique

Le problème du gradient de fuite a été découvert pour la première fois par Sepp (Joseph) Hochreiter en 1991. Sepp est un scientifique de génie et l'un des fondateurs, qui a contribué de manière significative à la façon dont nous utilisons les RNN et les LSTMs aujourd'hui. La deuxième personne est Yoshua Bengio, professeur à l'Université de Montréal. Il a également découvert le problème de la descente évanescente, mais un peu plus tard il a écrit à ce sujet en 1994. Yoshua est une autre personne qui repousse les limites dans l'espace de l'apprentissage en profondeur et des RNN, en particulier.

4.3.2 Expliquer le problème du gradient de fuite

L'algorithme de descente du gradient trouve le minimum global de la fonction de coût qui sera une configuration optimale pour le réseau, les informations voyagent à travers le réseau de neurones des neurones d'entrée aux neurones de sortie, tandis que l'erreur est calculée et propagée à travers le réseau pour mettre à jour les poids. Cela fonctionne de manière assez similaire pour les RNN, mais ici, nous avons un peu plus de choses à faire.

- Premièrement, les informations voyagent dans le temps dans les RNN, ce qui signifie que les informations des points de temps précédents sont utilisées comme entrée pour les points de temps suivants.
- Deuxièmement, vous pouvez calculer la fonction de coût, ou votre erreur, à chaque instant.

En gros, pendant la formation, votre fonction de coût compare vos résultats au résultat souhaité.

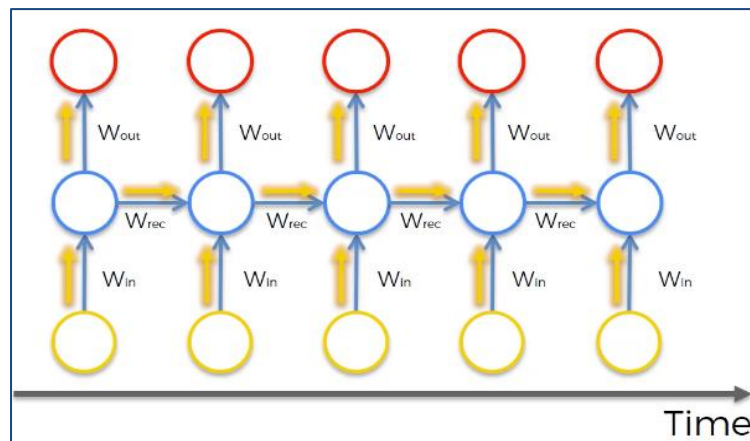


Figure 4-8 : Le voyage des observations dans un RNN.

La fonction de coût du réseau neuronal est calculée pour chaque observation dans l'ensemble de données. Ces valeurs de fonction de coût sont représentées en haut de l'image suivante :

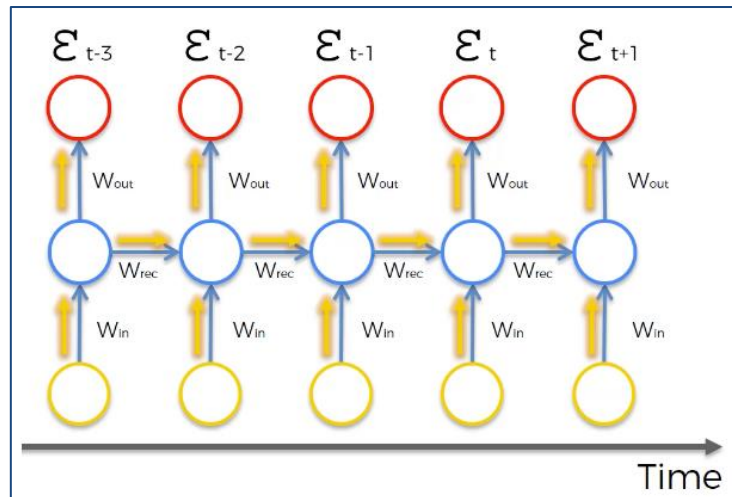


Figure 4-9 : Les résultats du voyage des observations dans les RNN.

Le problème du gradient de fuite se produit lorsque l'algorithme de rétro propagation revient à travers tous les neurones du réseau neuronal pour mettre à jour leurs poids. La nature des réseaux neuronaux récurrents signifie que la fonction de coût calculée au niveau d'une couche profonde du réseau neuronal sera utilisée pour modifier le poids des neurones au niveau des couches moins profondes.

Les mathématiques qui calculent ce changement sont multiplicatives, ce qui signifie que le gradient calculé dans une étape profonde dans le réseau de neurones sera multiplié par les poids plus tôt dans le réseau. Autrement dit, le gradient calculé en profondeur dans le réseau est "dilué" lorsqu'il revient à travers le réseau, ce qui peut faire disparaître le gradient - donnant le nom au problème du gradient de fuite !

Le facteur réel qui est multiplié par un réseau neuronal récurrent dans l'algorithme de rétro propagation est désigné par la variable mathématique W_{rec} . Cela pose deux problèmes :

- Lorsque W_{rec} est petit, vous rencontrez un problème de gradient de fuite
- Lorsque W_{rec} est grand, vous rencontrez un problème de gradient explosif

Notez que ces deux problèmes sont généralement désignés par le nom plus simple de "problème de gradient de fuite". [2.16].

4.3.3 Qu'est-ce que cela signifie pour le réseau ?

Plus le gradient est faible, plus il est difficile pour le réseau de mettre à jour les poids, et plus il faut du temps pour arriver au résultat final. Cependant, le problème n'est pas seulement que la moitié du réseau n'est pas entraîné correctement. La sortie des couches précédentes est utilisée comme entrée pour les couches suivantes. Ainsi, la formation pour le point temporel t se produit tout le temps sur la base d'entrées provenant de couches non formées. Ainsi, à cause du gradient de fuite, l'ensemble du réseau n'est pas entraîné correctement.

En résumé, le problème du gradient de fuite est causé par la nature multiplicative de l'algorithme de rétro propagation. Cela signifie que les gradients calculés à un stade profond du réseau neuronal récurrent ont un impact trop faible (dans un problème de

gradient de fuite) ou un impact trop important (dans un problème de gradient explosif) sur les poids des neurones qui sont moins profonds dans le réseau de neurones. Plus vous avancez dans le réseau, plus votre gradient est faible et plus il est difficile d'entraîner les poids, ce qui a un effet domino sur tous les autres poids du réseau. C'était le principal obstacle à l'utilisation des réseaux de neurones récurrents.

4.3.4 Solutions au problème du gradient de fuite

Bien sûr, cela ne sera pas un obstacle car ce problème peut être résolu comme suit :

En cas de gradient qui explose, vous pouvez :

- arrêter la rétro propagation après un certain point, ce qui n'est généralement pas optimal car tous les poids ne sont pas mis à jour ;
- pénaliser ou réduire artificiellement la pente ;
- mettre une limite maximale sur un gradient.

En cas de dégradé évanescent, vous pouvez :

- initialiser les poids de sorte que le potentiel du gradient de fuite soit minimisé ;
- avoir des réseaux d'état d'écho conçus pour résoudre le problème du gradient de fuite ;
- avoir des réseaux de mémoire longue à court terme (LSTM).

Les LSTMs sont considérés comme le réseau incontournable pour la mise en œuvre des RNN.

4.3.5 Problème du gradient de fuite et le LSTM

Le réseau de mémoire longue à court terme (LSTM) est la solution la plus populaire au problème du gradient de fuite.

Les LSTMs ont été créés pour traiter le problème du gradient de fuite. Alors, faisons un bref rappel sur cette question. Au fur et à mesure que nous propageons l'erreur à travers le réseau, elle doit passer par la boucle temporelle démêlée, les couches cachées reliées à elles-mêmes dans le temps au moyen de poids W_{rec} . Étant donné que ce poids est appliqué de nombreuses fois sur lui-même, le gradient diminue rapidement. Par conséquent, les poids des calques tout à gauche sont mis à jour beaucoup plus lentement que les poids des calques tout à droite. Cela crée un effet domino car les poids des calques d'extrême gauche définissent les entrées des calques d'extrême droite. Par conséquent, toute la formation du réseau en souffre, et c'est ce qu'on appelle le problème du gradient de fuite.

Nous avons également défini qu'en règle générale, si W_{rec} est petit (le gradient disparaît, et si W_{rec} est grand) le gradient explose. Mais qu'est-ce qui est « grand » et « petit » dans ce contexte ?

En fait, on peut dire que l'on a un gradient nul si $W_{rec} < 1$ et un gradient explosif si $W_{rec} > 1$. Alors, quelle est la première chose qui vous vient à l'esprit pour résoudre ce problème ?

Probablement, la solution la plus simple et la plus rapide sera de faire $W_{rec} = 1$. C'est exactement ce qui a été fait dans les LSTM. Bien sûr, il s'agit d'une explication très simplifiée, mais en général, rendre le poids récurrent égal à 1 est l'idée principale derrière les LSTM.

4.4 Mémoire à long court terme (LSTM)

Les réseaux de mémoire à long terme - généralement simplement appelés "LSTM" - sont un type spécial de RNN, capable d'apprendre des dépendances à long terme. Ils ont été introduits par Hochreiter & Schmidhuber (1997), et ont été raffinés et popularisés par de nombreuses personnes; Ils fonctionnent extrêmement bien sur une grande variété de problèmes, et sont maintenant largement utilisés.

Les LSTMs sont explicitement conçus pour éviter le problème de dépendance à long terme. Se souvenir d'informations pendant de longues périodes est pratiquement leur comportement par défaut, pas quelque chose qu'ils ont du mal à apprendre !.

4.4.1 Architecture LSTM

L'image suivante montre à quoi ressemblent les LSTMs :

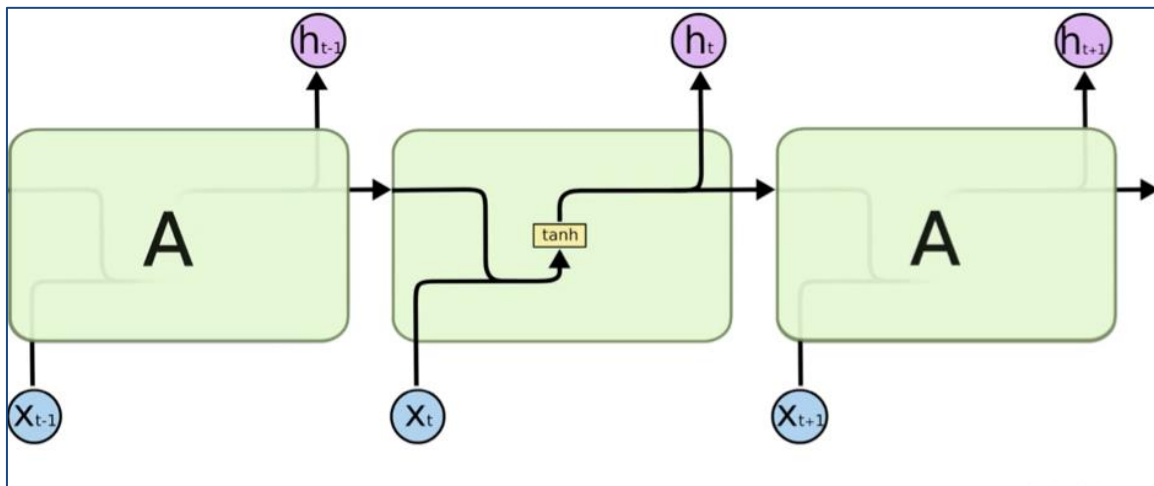


Figure 4-10 : Module répétitif dans un RNN standard contient une seule couche.

Les LSTMs ont également cette structure en forme de chaîne, mais le module répétitif à une structure différente.

Au lieu d'avoir une seule couche de réseau de neurones, il y en a quatre, qui interagissent d'une manière très spéciale.

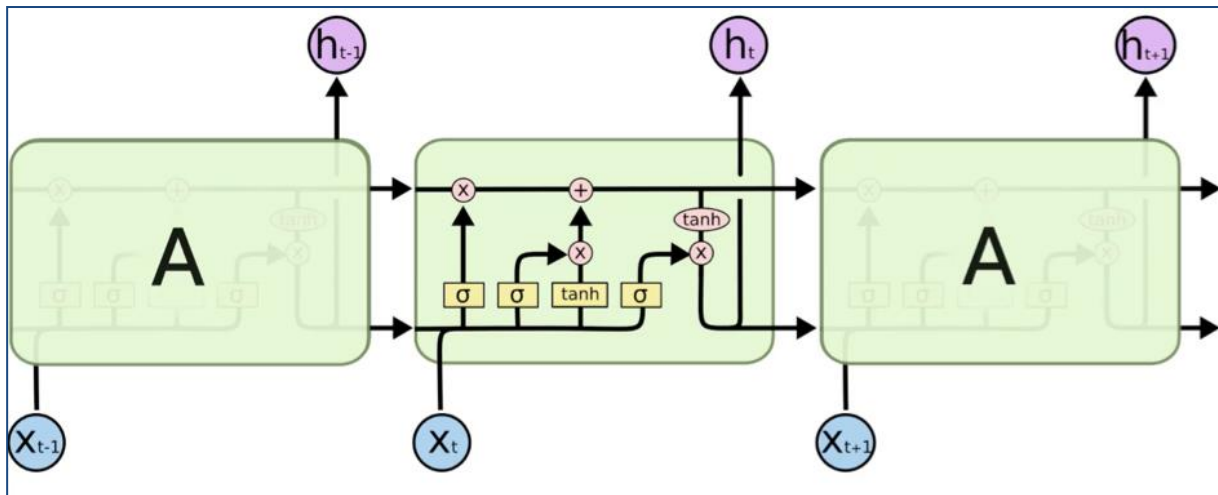


Figure 4-11 : Module répétitif dans un LSTM standard contient quatre couches.

- c_{t-1} représente l'entrée d'une cellule de mémoire à l'instant t ;
- x_t est une entrée à l'instant t ;
- h_t est une sortie au point temporel t qui va à la fois à la couche de sortie et à la couche cachée au point temporel suivant.

Ainsi, chaque bloc a trois entrées (x_t, h_{t-1} etc c_{t-1}) et deux sorties (h_t etc c_t). Une chose importante à retenir est que toutes ces entrées et sorties ne sont pas des valeurs uniques, mais des vecteurs avec beaucoup de valeurs derrière chacune d'elles.

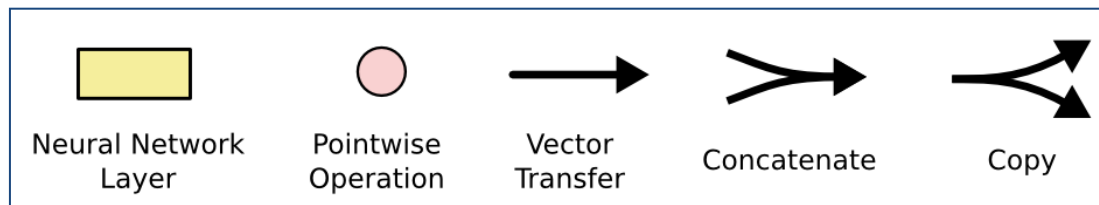


Figure 4-12 : Représentation plus approchée des continus de la Figure 4-11.

Dans le diagramme ci-dessus, chaque ligne porte un vecteur entier, de la sortie d'un nœud aux entrées des autres. Les cercles roses représentent des opérations ponctuelles, comme l'addition de vecteurs, tandis que les cases jaunes sont des couches de réseau neuronal apprises. La fusion de lignes indique une concaténation, tandis qu'une bifurcation de ligne indique que son contenu est copié et que les copies vont à des emplacements différents.

4.4.2 Procédure pas à pas de LSTM

Nous sommes maintenant prêts à examiner l'architecture LSTM étape par étape :

- 1) Nous avons la nouvelle valeur x_t et la valeur du nœud précédent h_{t-1} qui arrivent.

2) Ces valeurs sont combinées et passent par la fonction d'activation sigmoïde, où il est décidé si la valve d'oubli doit être ouverte, fermée ou ouverte dans une certaine mesure.

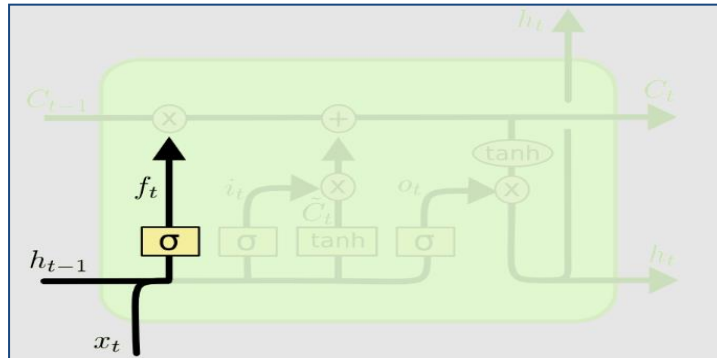


Figure 4-13 : Porte d'oubli.

Avec f_t donner comme suivant : $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$ (4.4)

- ✓ h_{t-1} : La sortie à l'instant t-1 ;
- ✓ x_t : L'entrée courante à l'instant t ;
- ✓ b_f : Un vecteur de biais ;
- ✓ w_f : Une matrice de poids ;
- ✓ σ : Fonction sigmoïde utilisée.

3) Les mêmes valeurs, ou en fait des vecteurs de valeurs, passent en parallèle par une autre opération de couche \tanh où il est décidé quelle valeur nous allons passer au pipeline de mémoire, et aussi une opération de couche sigmoïde, où il est décidé, si cette valeur va être transmise au pipeline de mémoire et dans quelle mesure.

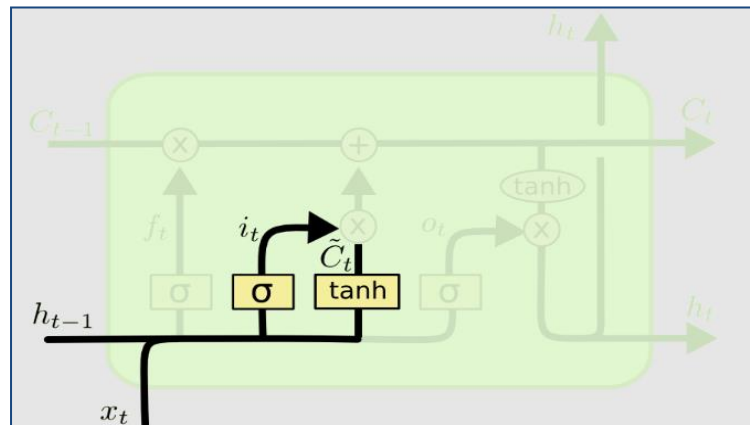


Figure 4-14 : Mise à jour de la cellule.

Avec $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ (4.5)

Et $\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$ (4.6)

- ✓ \tanh : La fonction d'activation tangente hyperbolique ;
- ✓ \tilde{c}_t : Valeur candidate.

4) Ensuite, nous avons une mémoire circulant dans le pipeline supérieur. Si nous avons oublié la valve ouverte et la valve de mémoire fermée, la mémoire ne changera pas. Sinon, si nous avons oublié la vanne fermée et la vanne mémoire ouverte, la mémoire sera complètement mise à jour.

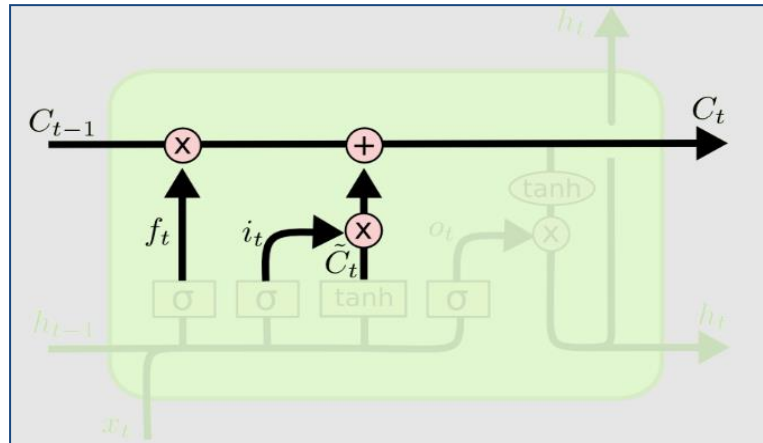


Figure 4-15 : Mise à jour de l'ancien état de cellule.

Avec : $c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$ (4.7)

- ✓ c_t : État interne

5) Enfin, nous avons combiné x_t et h_{t-1} pour décider quelle partie du pipeline de mémoire va devenir la sortie de ce module.

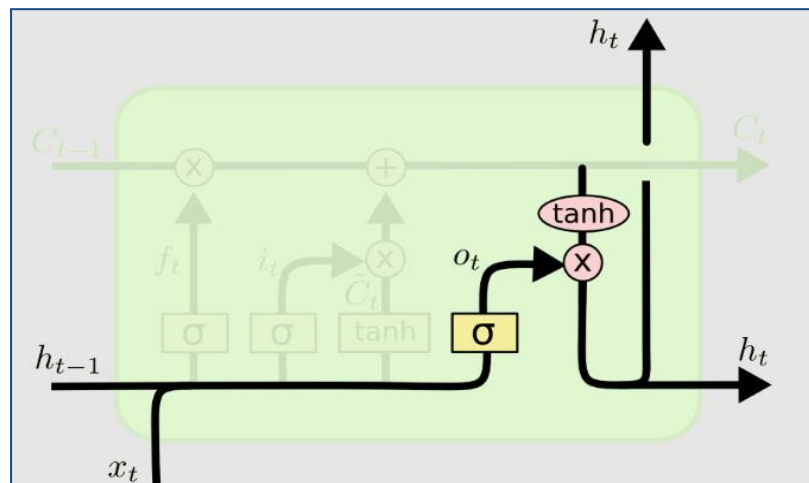


Figure 4-16 : La sortie de réseau LSTM.

Avec : $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ (4.8)

Et : $h_t = o_t * \tanh(c_t)$ (4.9)

✓ h_t : Sortie obtenue.

Ces cinq étapes sont essentiellement ce qui se passe au sein du réseau LSTM.

4.4.3 Variante LSTM

Ce qu'on a présenté jusqu'à présent est un LSTM assez normal. Mais tous les LSTMs ne sont pas identiques à ceux mentionnés ci-dessus. En fait, il semble que presque tous les articles impliquant des LSTMs utilisent une version légèrement différente. Les différences sont mineures, mais il convient d'en mentionner quelques-unes.

Voici le LSTM standard :

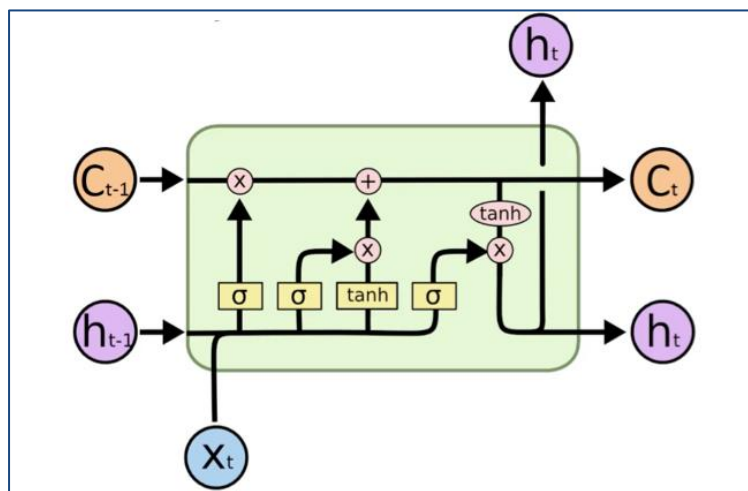


Figure 4-17 : Un LSTM standard.

Voyons maintenant quelques variantes.

Variante #1 :

La variante n°1, est une variante LSTM populaire, introduite par Gers & Schmidhuber (2000), ajoute des "connexions judas". Cela signifie que nous laissons les couches de grille regarder l'état de la cellule (les lignes qui alimentent les informations supplémentaires sur l'état actuel de la cellule mémoire aux fonctions d'activation sigmoïde).

$$f_t = \sigma(W_f \cdot [c_{t-1}, h_{t-1}, x_t] + b_f) \quad (4.10)$$

$$i_t = \sigma(W_i \cdot [c_{t-1}, h_{t-1}, x_t] + b_i) \quad (4.11)$$

$$o_t = \sigma(W_o \cdot [c_{t-1}, h_{t-1}, x_t] + b_o) \quad (4.12)$$

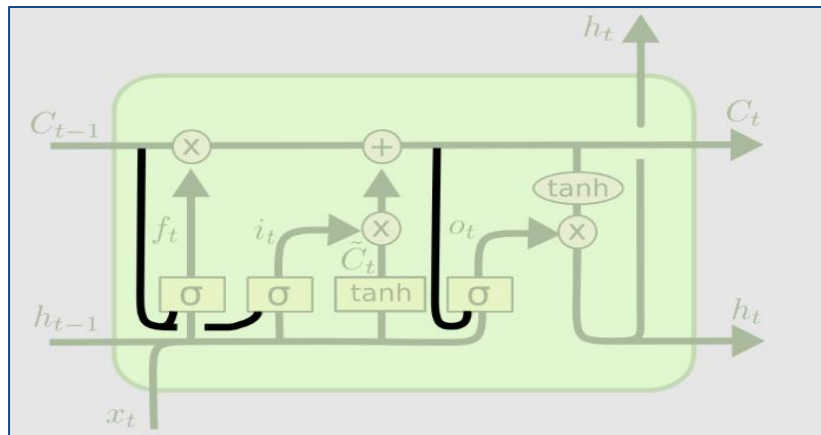


Figure 4-18 : La variante n°1 du LSTM.

Variante #2 :

Dans la variante n ° 2, nous connectons la valve d'oubli et la valve de mémoire. Ainsi, au lieu d'avoir des décisions séparées sur l'ouverture et la fermeture des vannes d'oubli et de mémoire, nous avons ici une décision combinée. Fondamentalement, chaque fois que vous fermez la mémoire (valve oubliée = 0), vous devez mettre quelque chose dedans (valve de mémoire = 1 - 0 = 1), et vice versa.

$$c_t = f_t * c_{t-1} + (1 - f_t) * \tilde{c}_t \quad (4.13)$$

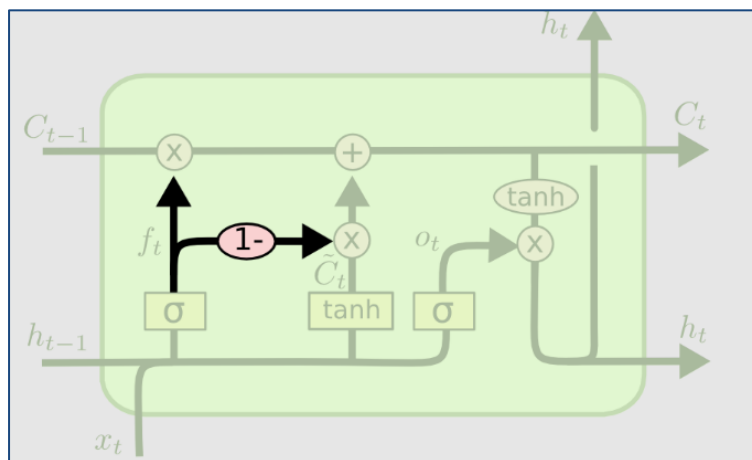


Figure 4-19 : La variante n°2 du LSTM.

Variante #3 :

La variante n ° 3 est généralement appelée unité récurrente fermée (GRU). Cette modification supprime complètement la cellule mémoire et la remplace par le pipeline caché. Ainsi, ici au lieu d'avoir deux valeurs distinctes - une pour la mémoire et une pour l'état caché- vous n'avez qu'une seule valeur.

Cela peut sembler assez complexe, mais en fait, le modèle résultant est plus simple que le LSTM standard. C'est pourquoi cette modification devient de plus en plus populaire.

Avec :

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (4.14)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (4.15)$$

$$\tilde{h}_t = \tanh(w \cdot [r_t * h_{t-1}, x_t]) \quad (4.16)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (4.17)$$

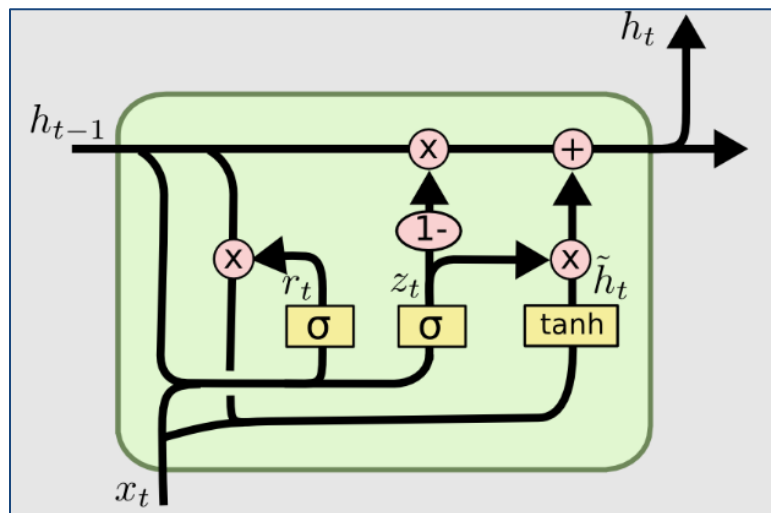


Figure 4-20 : La variante n°3 du LSTM

Ce ne sont que quelques-unes des variantes LSTM les plus notables. Il y en a beaucoup d'autres, comme les RNN Depth Gated de Yao.

Il existe également une approche complètement différente pour lutter contre les dépendances à long terme, comme les RNN Clockwork de Koutnik.

Laquelle de ces variantes est la meilleure ? Les différences importent-elles ?

Greff et al (2015) font une belle comparaison des variantes populaires, constatant qu'elles sont tous à peu près identiques. Jozefowicz a testé plus de dix mille architectures RNN, en trouvant certaines qui fonctionnaient mieux que les LSTMs sur certaines tâches [3.9].

4.5 Conclusion

Le LSTM a été inventé pour résoudre le problème du vanishing and exploding gradient rencontré dans un réseau de neurones récurrent classique.

Bref, LSTM est une topologie neuronale extrêmement utile à partir du moment où une "série temporelle" de données est en jeu.

Chapitre 5 : Application

*« Vous riez d'une prédiction sinistre et invraisemblable
Vous rirez moins si cette prédiction s'accomplit en partie
Le plus courageux des hommes attendra alors la suite ».*

EMILE AUGUSTE CHARTIER

5.1 Introduction

L'utilisation des méthodes de prévision a connu une évolution importante. Les chercheurs et praticiens ont eu de plus en plus recours aux nouvelles techniques jugées performantes.

Les réseaux de neurones constituent l'un des instruments les plus privilégiés pour reconnaître la prédiction des cours boursiers.

Dans ce chapitre on va exécuter deux codes python de prédiction « ANN & LSTM» sur cinq cours boursiers.

5.2 Préparation pour exécution

On a choisi de faire notre étude sur les cinq cours boursiers suivants :

__ Cours de blé : Le blé fait partie des trois grandes céréales avec le maïs et le riz.

__ Cours de l'or : De tous les métaux précieux, l'or est le plus populaire comme investissement.

__ Cours de pétrole : le pétrole est une importante source d'énergie primaire c'est pourquoi il est appelé or noire en raison de sa grande importance économique.

__ Cours de l'action Google : Google est définitivement une entreprise passionnante travaillant dans une industrie prometteuse.

__ Cours de l'action Boeing : Boeing est un leader mondial des industries aérospatiales et le plus grand fabricant d'avions commerciaux et militaires, de systèmes de sécurité et d'aérospatiale.

Nous avons téléchargé les données historiques de ces cours boursiers sur le site *yahoo finance* et sélectionné des données, puis enregistré ces données sous Excel.

Tableau 5-1 : Information des cinq actions boursières.

	<i>Données teste sélectionnés</i>	<i>Date du _ au _</i>	<i>Données training sélectionnés</i>	<i>Date du _ au _</i>
Cours de blé	<i>114 données</i>	<i>Du 03-01-2022 au 28-01-2022</i>	<i>7548 données</i>	<i>Du 03-01-2017 au 30-12-2021</i>
Cours de l'or	<i>114 données</i>	<i>Du 03-01-2022 au 28-01-2022</i>	<i>5028 données</i>	<i>Du 04-09-2018 au 30-12-2021</i>
Cours de pétrole	<i>114 données</i>	<i>Du 03-01-2022 au 28-01-2022</i>	<i>7548 données</i>	<i>Du 03-01-2017 au 30-12-2021</i>
Cours de l'action Google	<i>100 données</i>	<i>Du 03-01-2017 au 31-01-2017</i>	<i>6290 données</i>	<i>Du 01-03-2012 au 30-12-2016</i>
Cours de l'action Boeing	<i>114 données</i>	<i>Du 03-01-2022 au 28-01-2022</i>	<i>7548 données</i>	<i>Du 03-01-2017 au 30-12-2021</i>

Chacun de ces cours citées à six valeurs (prix) qui seront des entrées pour notre réseau. (*open/ high/ low/ close/ adj close/ volume*).

- ✓ Les données training ont pour but d'entraîner le programme (le réseau) afin d'analyser les informations et relier le comportement de ces actions.
- ✓ C'est sur les données test que se fera notre prédiction.

5.3 Présentation des résultats des programmes

Après avoir étudié le cours des actions pour chacun des marchés de matériaux ci-dessus et en utilisant l'algorithme ANN et LSTM, on a obtenu un ensemble de résultats sous forme graphique.

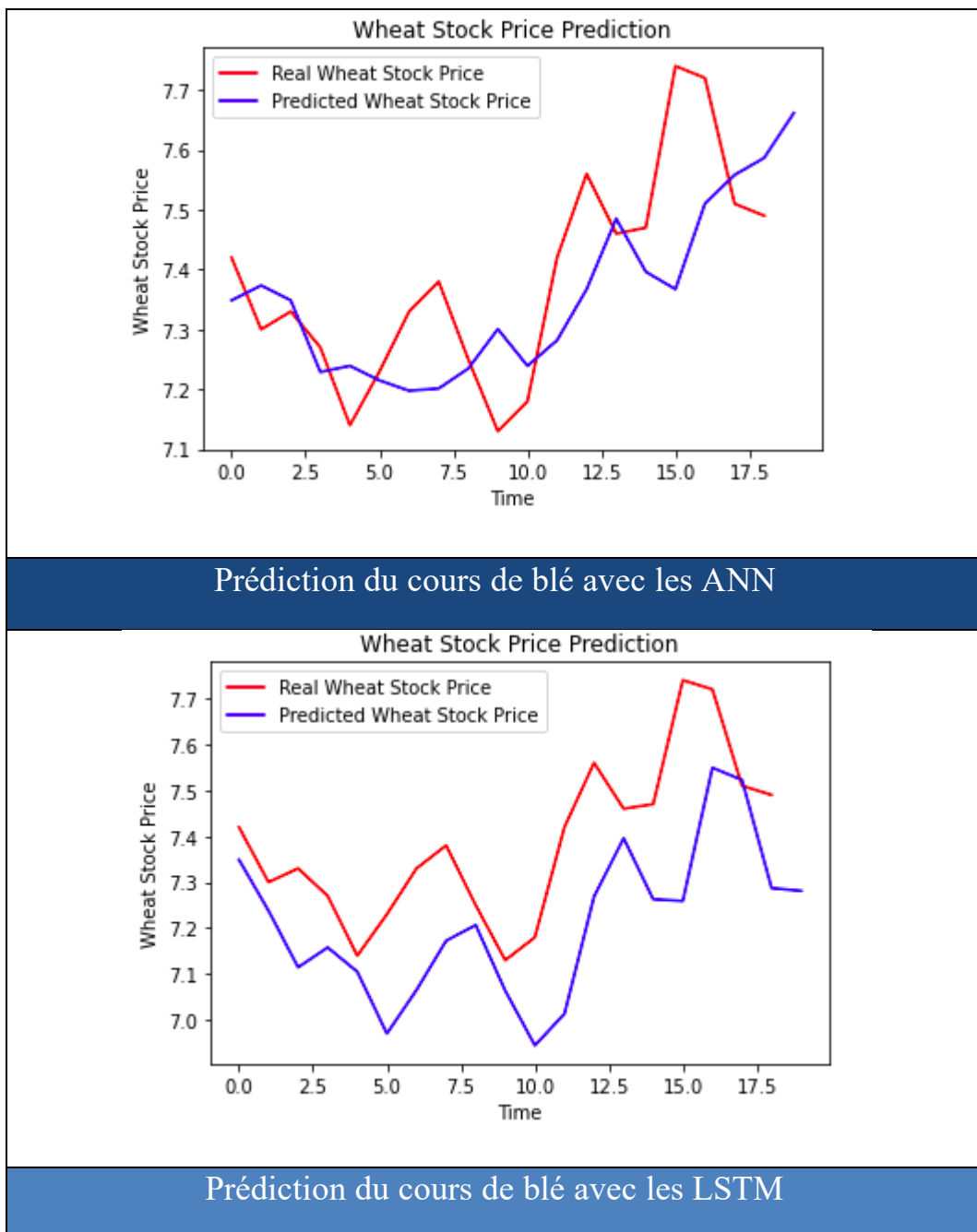
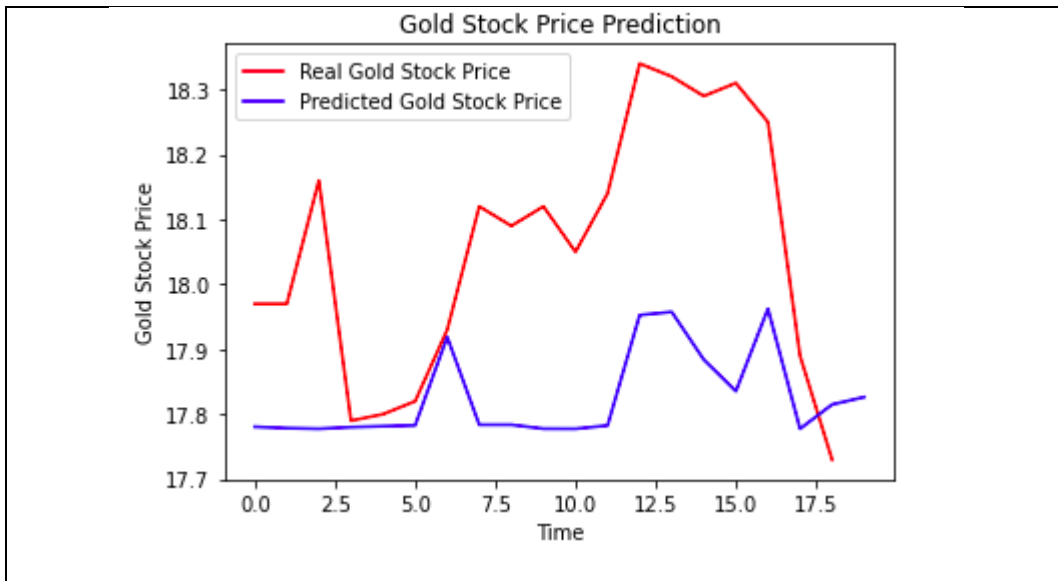
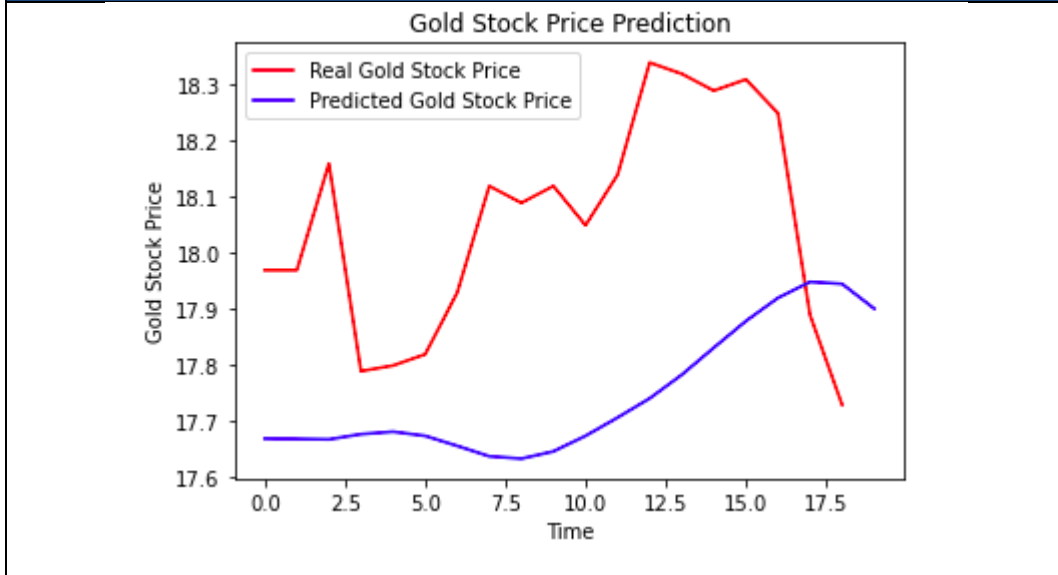


Figure 5-1 : Prédiction du cours de blé.

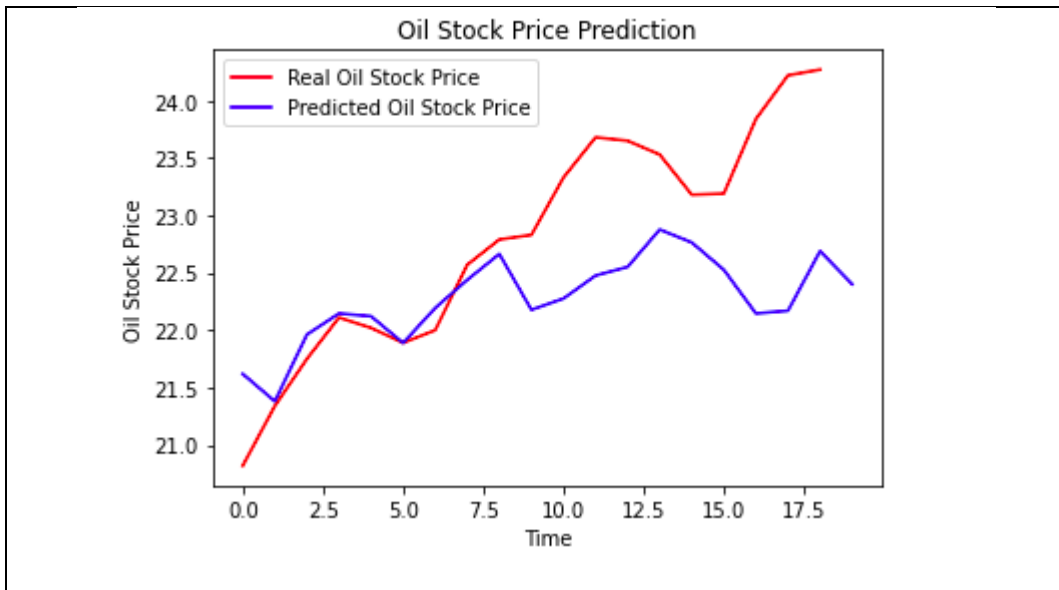


Prédiction du cours de l'or avec les ANN

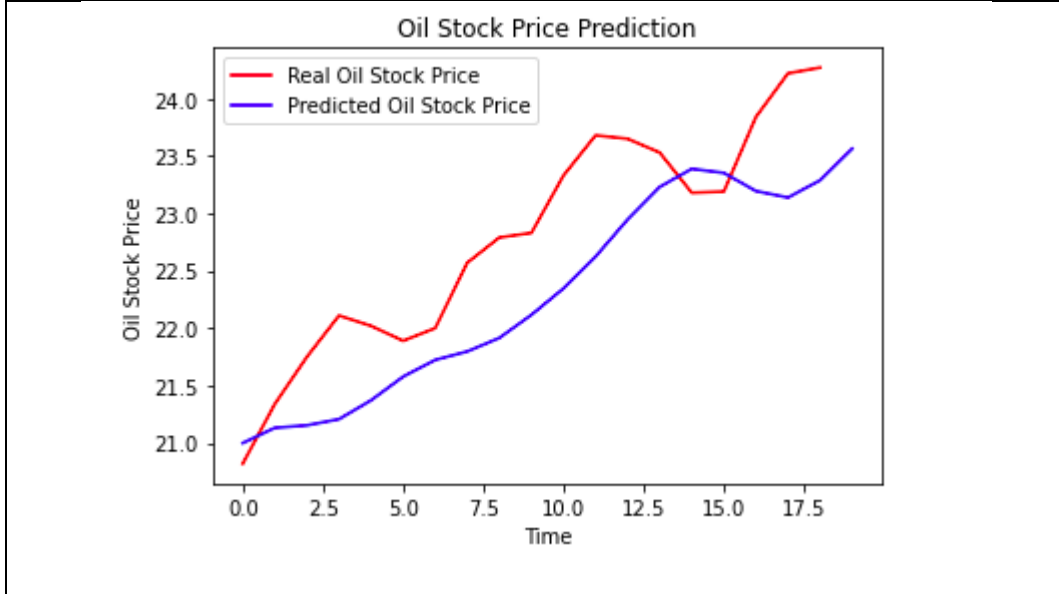


Prédiction du cours de l'or avec les LSTM

Figure 5-2 : Prédiction du cours de l'or.

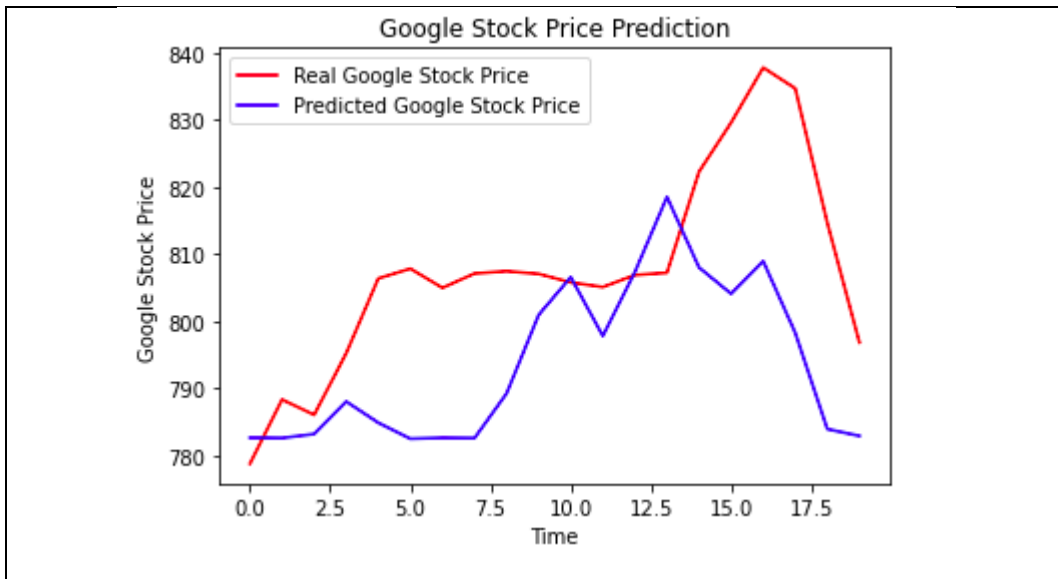


Prédiction du cours de pétrole avec les ANN

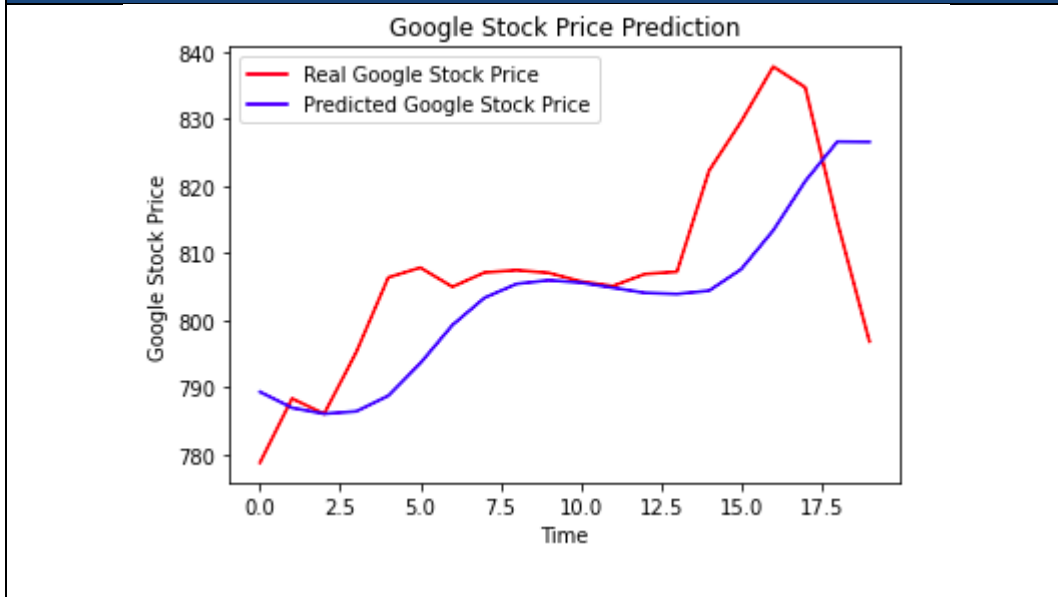


Prédiction du cours de pétrole avec les LSTM

Figure 5-3 : Prédiction du cours de pétrole.

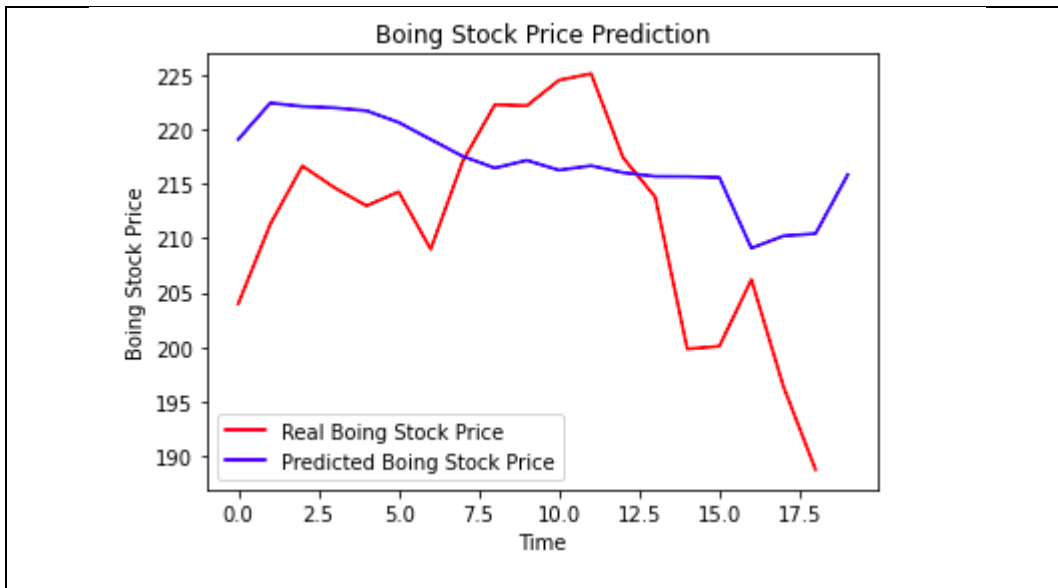


Prédiction du cours de l'action Google avec les ANN

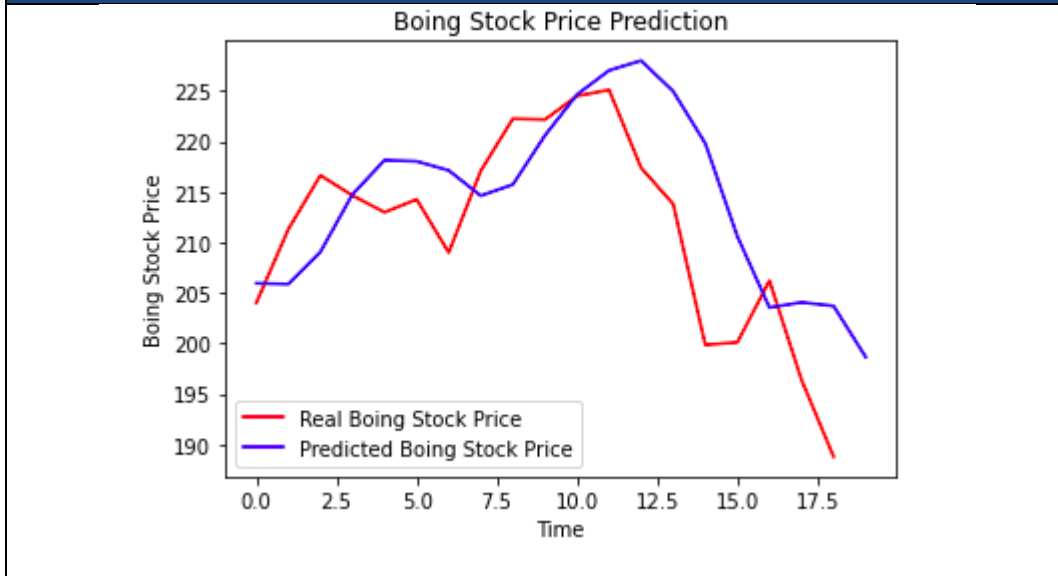


Prédiction du cours de l'action Google avec les LSTM

Figure 5-4 : Prédiction du cours de l'action Google.



Prédiction du cours de l'action Boeing avec les ANN



Prédiction du cours de l'action Boeing avec les LSTM

Figure 5-5 : Prédiction du cours de l'action Boeing.

En comparant les résultats obtenus de l'exécution des deux algorithmes de prédiction; on conclut que la prédiction est meilleure avec les LSTM et plus proche des prix réels contrairement à l'ANN.

Comme la note graphique est insuffisante, on fait quelques calculs représentés dans les tableaux que l'on présente :

Tableau 5-2 : Erreur moyenne (entre les vrais (prix) et les valeurs (prix) prédites).

	<i>Erreur moyenne ANN</i>	<i>Erreur moyenne LSTM</i>
<i>cours de blé</i>	0.90	0.10
<i>cours de l'or</i>	0.23	0.24
<i>cours de pétrole</i>	0.63	0.50
<i>cours de l'action Google</i>	15.08	10.73
<i>cours de l'action Boeing</i>	07.56	07.34

Les calculs du tableau sont faits en utilisant la formule suivant :

$$\boxed{\text{Erreur moyenne} = \text{moyenne}|\text{vrai prix} - \text{prix prédits}|}$$

Le tableau ci-dessus nous fournit des calculs d'erreurs pour les cinq cours boursiers de notre étude.

On générale, l'erreur moyenne commise avec les LSTMs et moins remarquable que celle avec les ANN.

$$\text{Erreur moyenne ANN} > \text{Erreur moyenne LSTM}$$

Cela confirme nos remarques graphiques.

Pour avoir un résultat qui valide notre point de vue, faisons un calcul final :

Tableau 5-3 : Erreur moyenne des cours entre les ANN et LSTM.

	<i>Total de l'erreur moyenne des cinq cours boursiers (TEMCB)</i>	<i>Moyenne de l'erreur des cinq cours boursiers (MECB)</i>
<i>ANN</i>	24.4	4.88
<i>LSTM</i>	18.91	3.782

Avec $\boxed{MECB = \frac{TEMCB}{5}}$ et $\boxed{TEMCB = \sum_{i=1}^5 \text{Erreur moyenne}}$.

5.4 Conclusion

De cette étude de comparaison entre les ANN et LSTM, on déduit que la prédiction avec les LSTMs donne des résultats plus fiables.

Conclusion générale

L'objectif de notre travail est d'explorer et d'étudier l'algorithme LSTM ainsi que son rôle pour la prédiction des cours boursiers. Nous avons divisé notre mémoire en deux grandes parties : partie théorique et partie pratique.

Dans la partie théorique, nous avons d'abord fait appel au machine Learning. Ensuite on a présenté les introductions et les notions de base des réseaux de neurones. Nous avons pris en particulier les réseaux de neurones récurrents ou les LSTMs qui représentent un cas particulier dans ce dernier. On a exposé les concepts ainsi que les notions de base des LSTMs puis on est passé à l'étude de son fonctionnement.

Dans la partie pratique, on a étudié une implémentation élémentaire des algorithmes LSTM et ANN, où les principales fonctions sont illustrées dans un code Python. Ainsi nous avons ajouté des captures d'écran pour présenter les résultats sous forme graphique. Ces graphes décrivent la prédiction des cours boursiers par les algorithmes étudiés. Pour valider l'hypothèse énoncée dans l'introduction, on a fait une interprétation de ces résultats en plus des calculs d'erreurs.

Finalement, nous sommes arrivés à une conclusion, qui est la suivante : LSTM peut effectivement faire des prévisions futures satisfaisantes et efficaces à la fois pour les stocks.

Ces réseaux de neurones récurrents LSTM « mémoire court-terme » ont révolutionné la reconnaissance de la voix (Speech Recognition), la compréhension et la génération de texte (Natural Language Processing) ou la prédiction sur des séries numériques de données temporelles. Il est clair que pour la prédiction de séries chronologiques, les LSTMs donnent de bien meilleurs résultats que les réseaux de neurones traditionnels qui ne sont pas adaptés à ce type de tâche. Le temps de calcul en phase d'apprentissage est cependant nettement augmenté avec les LSTMs du fait d'un nombre de poids du réseau fortement augmenté.

Ce n'est peut-être pas le meilleur moyen de prédire l'évolution des cours boursiers surtout avec le développement technologique que connaît notre époque. Mais l'architecture actuellement utilisée peut ne pas être adaptée aux problèmes actuels, il reste à dire que ces résultats peuvent être efficaces et avec un taux d'erreur moyen.

Annexe pour les algorithmes

```
///ANN///

# -*- coding: utf-8 -*-
"""Ann.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1vVDFrx1jNqqZktLp1Ja9RAaylGJ0xGRB
"""

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset_train = pd.read_csv('Google_Stock_Price_Train.csv')
training_set = dataset_train.iloc[:, 1:2].values

training_set

from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)

#o_min = open.min()
#normalized_o = (open - o_min) / (open.max() - o_min)
training_set_scaled

from keras.preprocessing.sequence import TimeseriesGenerator

tsg = TimeseriesGenerator(training_set_scaled, training_set_scaled, 60,
batch_size=len(training_set_scaled))

X_train, Y_train = tsg[0]

X_train

Y_train

from keras.models import Sequential
from keras.layers import Dense
#from keras.layers import LSTM
from keras.layers import Dropout

X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))

from keras import layers
regressor = Sequential()
```

```

regressor.add(layers.Dense(50, activation='relu', input_shape =
(X_train.shape[1], 1)))
#regressor.add(Dropout(0.2))
regressor.add(layers.Dense(50, activation='relu'))
#regressor.add(Dropout(0.2))

regressor.add(layers.Dense(50, activation='relu'))
#regressor.add(Dropout(0.2))
regressor.add(layers.Dense(50))
#regressor.add(Dropout(0.2))
regressor.add(layers.Dense(1))
#regressor.add(layers.Dense(50, activation='relu'))
#regressor.add(50, return_sequences = True, input_shape = (i.shape[1],
1))
#regressor.add(Dropout(0.2))
#regressor.add(units = 50, return_sequences = True)
#regressor.add(Dropout(0.2))
#regressor.add(units = 50, return_sequences = True)
#regressor.add(Dropout(0.2))
#regressor.add(units = 50)
#regressor.add(Dropout(0.2))
#regressor.add(Dense(units = 1))
    #m.add(layers.Dense(64, activation='relu', input_shape=(n+1,)))
    #m.add(layers.Dense(64, activation='relu'))
    #m.add(layers.Dense(1))
    #return m

regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')

h = regressor.fit(X_train, Y_train, epochs = 100, batch_size = 32)
        #epochs=epochs,
        #batch_size=32,
        #validation_data=(val_inputs, val_targets))

dataset_test = pd.read_csv('Google_Stock_Price_Test.csv')
real_stock_price = dataset_test.iloc[:, 1:2].values
#real_stock_price
dataset_test.shape

from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
inputs = sc.fit_transform(real_stock_price)
inputs.shape

dataset_total = pd.concat((dataset_train['Open'],
dataset_test['Open']), axis = 0)
inputs = dataset_total[len(dataset_total) - len(dataset_test) -
60:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)
X_test = []
for i in range(60, 80):
    X_test.append(inputs[i-60:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
predicted_stock_price = regressor.predict(X_test)

```

```
predicted_stock_price =
sc.inverse_transform(predicted_stock_price[:,0,:])predicted_stock_price
.shape

real_stock_price.shape

predicted_stock_price.shape

sum(abs(real_stock_price-predicted_stock_price[:
-1]))/len(predicted_stock_price)

plt.plot(real_stock_price, color = 'red', label = 'Real Oil Stock
Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Oil
Stock Price')
plt.title('Oil Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Oil Stock Price')
plt.legend()
plt.show()
```

```

///LSTM ///
# -*- coding: utf-8 -*-
"""Copie de rnn.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1Cs2GKZmmB-
sRZLk6HScT4kOXlK4Mzbxv

# Recurrent Neural Network

## Part 1 - Data Preprocessing

### Importing the libraries
"""

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

"""### Importing the training set"""

dataset_train = pd.read_csv('Google_Stock_Price_Train.csv')
training_set = dataset_train.iloc[:, 1:2].values

dataset_train

"""### Feature Scaling"""

from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)

"""### Creating a data structure with 60 timesteps and 1 output"""

X_train = []
y_train = []
#for i in range(60, 1258):
for i in range(60, 1258):
    X_train.append(training_set_scaled[i-60:i, 0])
    y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)

"""### Reshaping"""

X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))

"""### Part 2 - Building and Training the RNN

### Importing the Keras libraries and packages
"""

from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout

```

```

"""### Initialising the RNN"""
regressor = Sequential()

"""### Adding the first LSTM layer and some Dropout regularisation"""
regressor.add(LSTM(units = 50, return_sequences = True, input_shape =
(X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

"""### Adding a second LSTM layer and some Dropout regularisation"""
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

"""### Adding a third LSTM layer and some Dropout regularisation"""
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

"""### Adding a fourth LSTM layer and some Dropout regularisation"""
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))

"""### Adding the output layer"""
regressor.add(Dense(units = 1))

"""### Compiling the RNN"""
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')

"""### Fitting the RNN to the Training set"""
regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)

"""### Part 3 - Making the predictions and visualising the results

### Getting the real stock price of 2017
"""
dataset_test = pd.read_csv('Google_Stock_Price_Test.csv')
real_stock_price = dataset_test.iloc[:, 1:2].values

real_stock_price

"""### Getting the predicted stock price of 2017"""

dataset_total = pd.concat((dataset_train['Open'],
dataset_test['Open']), axis = 0)
inputs = dataset_total[len(dataset_total) - len(dataset_test) -
60:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)
X_test = []
for i in range(60, 80):
    X_test.append(inputs[i-60:i, 0])

```



```
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
predicted_stock_price = regressor.predict(X_test)
predicted_stock_price = sc.inverse_transform(predicted_stock_price)

"""### Visualising the results"""

X_test

sum(abs(real_stock_price-
predicted_stock_price))/len(predicted_stock_price)

plt.plot(real_stock_price, color = 'red', label = 'Real Oil Stock
Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Oil
Stock Price')
plt.title('Oil Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Oil Stock Price')
plt.legend()
plt.show()
```

Bibliographie

Livres :

- [1.1] Catherine Karyotis, *L'essentiel de la Bourse, France, Paris, Décembre 2008 (ISBN 978-2-297-01101-3)*
- [1.2] Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.
- [1.3] El Naqa, I., & Murphy, M. J. (2015). *What is machine learning? ; In machine learning in radiation oncology (pp. 3-11)*. Springer, Cham.
- [1.4] Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., ... & Zdeborová, L. (2019). *Machine learning and the physical sciences*. *Reviews of Modern Physics*, 91(4), 045002.
- [1.5] Claude. Touzet, *LES RESEAUX DE NEURONES ARTIFICIELS : INTRODUCTION AU CONNEXIONNISME*, 2016.
- [1.6] Dave Anderson and George McNeill, *ARTIFICIAL NEURALNETWORKS TECHNOLOGY*, Utica, New York, 1992.
- [1.7] Pierre Borne, Mohamed Benrejeb, Joseph Haggège, *les réseaux de neurones : présentation et application*. 2007.
- [1.8] Simon Haykin (16 juillet 1998). "Chapitre 2: Processus d'apprentissage". *Réseaux de neurones: une fondation complète (2e éd.)*. Prentice Hall. 50–104. ISBN 978-8178083001. Récupéré le 2 mai 2012.
- [1.9] S Russell, P. Norvig (1995). "Chapitre 18: Apprendre à partir d'exemples". *Intelligence artificielle: une approche moderne (3e éd.)*. Prentice Hall. 693–859. ISBN 0-13-103805-2. Récupéré le 20 novembre 2013
- [1.10] *Neural Networks and Deep Learning; Michael Nielsen; 2019.*

Thèses & ouvrages :

- [2.1] Lagneau-Ymonet, P., & Riva, A. (2015). *Histoire de la Bourse*. La Découverte.
- [2.2] *tout savoir sur les etf qu'est-ce qu'un etf ; étude BlackRock People & Money ; 2020.*
- [2.3] *Peut-on prédire le cours d'une action ? ; Guillaume Clément. Publié le 20/08/2019 ; Mis à jour le 20/08/2019.*
- [2.4] *siecldigital ; histoire intelligence artificielle; Valentin Blanchot; Publié le 20 août 2008.*

- [2.5] Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). *Machine learning on big data: Opportunities and challenges*. *Neurocomputing*, 237, 350-361
- [2.6] Ayodele, T. O. (2010). *Types of machine learning algorithms*. *New advances in machine learning*, 3, 19-48.
- [2.7] *Machine Learning : les 9 types d'algorithmes les plus pertinents en entreprise; lemagit ; Cassidy Kelley; Publié le 08 juin 2020.*
- [2.8] Antoine Crochet-Damais, (2022). *Le machine learning est une technique d'apprentissage automatique utilisée en intelligence artificielle*. JDN
- [2.9] Mackenzie, A. (2015). *The production of prediction: What does machine learning want?* *European Journal of Cultural Studies*, 18(4-5), 429-445.
- [2.10] De Paepe, A., Vrins, F., & Petitjean, M. *Etude des performances du Machine Learning dans la gestion de portefeuilles.*
- [2.11] *Mémoire de Magister ; Filtration des données sismiques par les réseaux de neurones artificiels ; Babaia Foudil ; boumerdes/ faculte de hydrocarbures et de la chimie ; 2015.*
- [2.12] *Mémoire de Magister ; optimisation de la structure des réseaux de neurones par algorithmes génétiques ; Houssou Mohammed ; 2005.*
- [2.13] *Mémoire de Magister ; application des réseaux de neurones pour la prédiction de la propagation des rotules plastiques ; Rafik Taleb ; 2005.*
- [2.14] *Une introduction douce à la fonction sigmoïde; NEURONE CONNECTION; Actualités de l'Intelligence Artificielle - Machine Learning - Objets connectés.*
- [2.15] *Machine Learning : Notions Fondamentales; Ahmed CHIKHAOUI; Mars 2018; Département de l'Informatique; Université Ibn Khaldoune Tيارت; Algérie.*
- [2.16] *Nick McCullum : Développeur de logiciels et explicateur professionnel ; Le problème du gradient de disparition dans les réseaux de neurones récurrents.*

Sites internet :

- [3.1] *Bourse définition traduction. Mis à jour le 02/02/19 09:01. Accède le 10/05/22. <https://www.journaldunet.fr/business/dictionnaire-economique-et-financier>.*
- [3.2] *Le rôle de la Bourse. Accède le 10/05/22. <https://www.zonebourse.com>.*
- [3.3] *Bourse. Accède le 11/05/22. <https://fr.wikipedia.org>.*
- [3.4] *Comment fonctionne la bourse. Accède le 15/05/22. <https://www.disnat.com>.*

- [3.5] *exchange traded-funds definition avantages et risque. Mise à jour le 09/03/21. Accède le 01/06/22.*
<https://www.journaldunet.fr/patrimoine/guide-des-finances-personnelles>.
- [3.6] *Une intelligence artificielle aurait-elle découvert le graal boursier?. Repéré par Ophélie Surcouf sur Fast Company le 25/05/2020. Accédé le 04/06/22.*
<https://korii.slate.fr/bi>.
- [3.7] *Deep Learning vs Machine Learning. Accède le 24/04/22.*
<https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/>.
- [3.8] *The Ultimate Guide to Recurrent Neural Networks (RNN). Published by Super Data Science Team. 08/30/2018. Accède le 27/05/22.*
<https://www.superdatascience.com/blogs/the-ultimate-guide-to-recurrent-neural-networks-rnn/>.
- [3.9] *Understanding LSTM Networks. Posted on August 27,2015. Accède le 13/06/22.*
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.